UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA


# RoboCISO: Um sistema de robôs de software (RPA) para alertar o CISO sobre o *status* dos seus vetores de risco mais críticos

**MESTRADO EM INFORMÁTICA**

**Cátia Sofia Silva Rodrigues**

Trabalho de projeto orientado por:
Profª Doutora Ana Luísa do Carmo Correia Respício

2020

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática


# RoboCISO: Um sistema de robôs de software (RPA) para alertar o CISO sobre o *status* dos seus vetores de risco mais críticos

**Cátia Sofia Silva Rodrigues**

**Trabalho de projeto**

MESTRADO EM INFORMÁTICA

Trabalho de projeto orientado pela Profª Doutora Ana Luísa do Carmo Correia Respício
e supervisionado na Altice pelo Eng. José António dos Santos Alegria

2020

# Agradecimentos

Ao longo deste percurso tive o apoio de muitas pessoas, esta secção serve como forma de agradecimento a todas elas.

Em primeiro lugar quero agradecer ao Eng. José Alegria pela orientação e ao Eng. Carlos Cabral por toda a ajuda ao longo do desenvolvimento deste projeto. Agradeço também ao restante departamento da DCY da Altice Portugal por me receber e por se disponibilizarem sempre que necessário.

Agradeço à minha orientadora, Professora Ana Respício pelo tempo disponibilizado e toda a orientação.

Quero agradecer aos meus colegas da MEO por todos os momentos que me proporcionaram ao longo deste trajeto. Obrigada à Sara Nascimento por toda ajuda no desenvolvimento do projeto e à Inês Rodrigues, José Águas, Beatriz Sécio, João Miranda e Gonçalo Miranda por todo o apoio e gargalhadas, fazendo assim os momentos mais difíceis passarem mais depressa.

Por último, quero agradecer aos meus pais, namorado, avós e irmão. Todos eles foram fundamentais para concluir este trabalho e sem eles nada disto era possível. Muito obrigada!

# Resumo

Um *Chief Security Information Officer* (CISO) lida todos os dias com uma grande quantidade de informação que deve ser cuidadosamente analisada de forma a prevenir a existência de incidentes de cibersegurança na sua organização.

O RoboCISO é um conjunto de processos robotizados, recorrendo à *Robotic Process Automation* (RPA), que visa auxiliar o CISO nas suas tarefas diárias de análise ao *status* dos seus vetores de risco mais críticos, distribuídos pelas vertentes de Urgência, Atenção e Rotina, sendo implementados cinco casos de uso nestas mesmas vertentes. Na vertente de Urgência, o primeiro caso de uso é focado nos alertas de ataques DDoS, tendo como objetivo diminuir a quantidade de alertas gerados. O segundo caso de uso foca-se na verificação do funcionamento das plataformas mais críticas. Se uma dada plataforma não estiver em funcionamento o CISO deverá ter essa informação, não esperando que esse alerta chegue por parte de terceiros. Na vertente de Atenção, o primeiro caso de uso é focado no *patching* de máquinas com sistema operativo Windows. No caso de existirem *patches* cuja instalação está em atrasado o CISO deverá estar a par dessa situação. O segundo caso de uso nesta vertente, aborda o *Service Level Agreement* (SLA) na resolução de incidentes. Se o tempo estipulado para resolução de incidentes não for cumprido, o CISO deve ser informado para que possa saber o porquê e tomar medidas para evitar que tal aconteça de novo. Na vertente de Rotina, o caso de uso diz respeito à verificação dos *ratings* da BitSight. O CISO analisa os *ratings* da Bitsight através de *heat maps*. Em caso de alterações acentuadas, deve ser gerado um alerta em tempo útil diminuindo a probabilidade de uma alteração significativa passar despercebida durante a análise dos *heat maps*.

O RoboCISO contribui desta forma para que o CISO esteja efetivamente informado sobre o estado dos seus vetores de risco mais críticos sem se preocupar com informação desnecessária e em excesso. Possuindo a informação situacional sobre cada vetor crítico, o CISO estará regularmente informado, e permitindo que em caso de incidentes de cibersegurança que exijam a sua atuação este possa agir de forma mais rápida e eficaz.

**Palavras-chave:** Alertas, DDoS, *Patching*, *Robotic Process Automation*, CISO

# Abstract

A Chief Security Information Officer (CISO) deals every day with a huge amount of information which needs to be carefully analyzed to prevent the occurrence of cyber incidents in the organization.

RoboCISO is a set of automated processes (RPA) that aims to assist the CISO in his daily tasks of analysis of the status of critical risk vectors. Five use cases were implemented focusing on the strands of Urgency, Attention and Routine. On the Urgency aspect, the first use case focuses on alerts of DDoS attacks and the objective is to decrease the number of alerts generated. The second use case focuses on checking if the most critical platforms are operating correctly. If any of the platforms checked isn't working properly the CISO must be informed as fast as possible so measures can be taken. In the Attention aspect, the first use case focuses on the delay of patching installation in machines with the Windows operative system. The second use case is related to the Service Level Agreement stipulated to resolve the different types of internal incidents. If the resolution time of some incident doesn't comply with the time agreed on, the CISO must be informed.Regarding the strand of Routine, the use case implemented focuses on the analysis of Bitsight ratings. The CISO analyzes Bitsight ratings through heat maps. In case of accentuated changes, a timely warning should be generated decreasing the likelihood of a significant change going unnoticed during the heat maps analysis.

RoboCISO allows the CISO to be effectively informed about the state of his risk vectors without worrying about unnecessary and excessive information. Having the situational information about each critical vector regularly, allows faster and more effective actions to be taken in the event of cyber security incidents.

**Keywords:** Alerts, DDoS, Patching, Robotic Process Automation, CISO

# Conteúdo

# Lista de Figuras

# Lista de Tabelas

# Capítulo 1

# Introdução

A MEO Serviços de Comunicações e Multimédia SA é uma empresa do grupo Altice Portugal e tem cerca de dez mil colaboradores e mais de três milhões de clientes, sendo a informação um ativo crítico essencial para o seu negócio. Dado que a informação é um ativo crítico a Altice segue os melhores métodos para proporcionar a sua segurança. A segurança de informação é assegurada através da implementação de um conjunto de controlos que inclui políticas, processos, procedimentos, estruturas organizadas, *software* e *hardware*. Estes controlos têm de ser estabelecidos, implementados, monitorizados, revistos e melhorados [1]. Num estudo conduzido pelo Penomen Institute a empresas de diversas áreas como a banca, comunicações e IT, 65% dos consumidores que sofreram uma *data breach* perderam a confiança na empresa e 27% descontinuou a relação com mesma [2].

A Direção de Cibersegurança e Privacidade (DCY) da Altice Portugal é responsável pela segurança da informação tendo o *Chief Information Security Officer* (CISO) no seu centro. O CISO é um gestor de nível executivo responsável por estabelecer e manter a visão, estratégia, operações e orçamento para a proteção dos ativos de informação e ativos informáticos da sua organização. O CISO assegura o cumprimento de todas as responsabilidades da direção e fazendo a administração perceber a importância da segurança de informação, negoceia de forma a obter o financiamento para tal. Este conjunto de ações designa-se por governança da cibersegurança [3], sendo a forma como esta é executada na Altice Portugal descrita na Secção 2.4.

O objetivo deste projeto é criar uma ferramenta de auxílio na execução de algumas das atividades diárias do CISO que lhe permita uma análise em tempo útil de alguns dos vetores que este considera como mais críticos à governança da cibersegurança.

## 1.1 Motivação

Todos os dias o CISO, para assegurar uma boa governança da cibersegurança, precisa de ter acesso em tempo útil a uma grande quantidade de informação, nomeadamente, indicadores do estado dos principais vetores de risco (*e.g.* ciber higiene). No entanto, nem toda essa informação é relevante em todos os momentos e o CISO corre o risco de *information/alert fatigue*. A motivação deste trabalho é a criação de um sistema que informe o CISO, em tempo útil, acerca do estado dos vetores de risco que este considera mais críticos, possibilitando uma melhor análise dos mesmos. Com um sistema com estas capacidades, o CISO estará informado sobre o estado dos vetores mais críticos ao funcionamento da infraestrutura.

Deste modo, o CISO poderá tomar decisões de forma atempada podendo mesmo evitar a ocorrência de incidentes de cibersegurança. Um desses incidentes que teve um impacto significativo a nível mundial afetando diversas organizações, entre elas a Altice Portugal [4] aconteceu a 12 de Maio de 2017 com o ciberataque de *ransomware*[1], conhecido como *WannaCry*. Este ataque consistiu na disseminação de um *software* malicioso, que cifra o disco de computadores com sistema operativo Windows afetando aproximadamente 2000 máquinas na infraestrutura da Altice Portugal. Isto deveu-se ao facto da política de gestão de *patches*[2] em curso, na altura do ataque, se basear na sua instalação após passarem por um processo de testes, que tinha o intuito de perceber quais as alterações que esses *patches* teriam nas aplicações utilizadas nas máquinas, em termos de desempenho, versões e funcionalidades. Esta fase de testes podia demorar meses passando muito tempo desde o lançamento do *patch* até à sua instalação e, consequentemente, muito tempo desde a divulgação pública das vulnerabilidades que esse *patch* mitiga. Desde então a política foi alterada estando brevemente descrita na Subseção 2.2.2.

Uma vez que, no início do projeto não existia nenhum processo que informasse o CISO em relação ao estado da instalação dos *patches* do sistema operativo Windows, este é um dos vetores críticos abordados neste projeto. Outro vetor crítico para o qual também não existia um processo que informasse o CISO em relação ao seu estado é o cumprimento dos *Service Level Agreements* (SLAs), descrito na Secção 1.2. O estado destes vetores deve ser reportado ao CISO com regularidade (*e.g.* uma vez por dia). Outros vetores críticos apresentam mais urgência devendo por isso gerar alertas, ou seja, sempre que existir uma situação crítica em curso o CISO deve ser alertado em tempo útil para essa situação. No início do projeto não existia nenhum processo que alertasse o CISO caso uma plataforma crítica deixasse de funcionar, como por exemplo a VPN, tendo essa informação de chegar via terceiros (*e.g.* a equipa SOC). Todos os ataques de DoS/DDoS sofridos pela organização geravam um alerta via SMS para o CISO. A maior parte destes ataques não possui uma dimensão significativa que justifique a notificação ao CISO o que originava um excesso de alertas.

Deste modo, a grande motivação deste trabalho foi a criação de um sistema que auxilie o CISO na execução da sua função, colmatando as lacunas existentes neste pequeno conjunto de vetores críticos.

## 1.2 Objetivos

O objetivo deste projeto é construir um sistema de processos automatizados, recorrendo a *Robotic Process Automation* (RPA), que se torne numa ferramenta de suporte à tomada de decisão do CISO. Esta ferramenta deve ser responsável por reportar em tempo útil o estado dos vetores de risco que o CISO considera mais críticos ao funcionamento da sua infraestrutura tornando-se assim um "assistente virtual" na governança da cibersegurança da Altice Portugal.

Dado o caráter urgente da informação que é transmitida ao CISO o sistema utiliza uma aplicação de mensagens instantâneas, para além do convencional *e-mail*, para o envio de notificações e alertas uma

[1] O *Ransomware* é uma forma de *malware* (*software* malicioso desenvolvido com o intuito de causar danos a dados, dispositivos ou pessoas) que consiste geralmente em negar o acesso dos utilizadores aos seus próprios dados, exigindo um "resgate" com a promessa de restauro de acesso aos dados mediante o pagamento. Esta forma de ataque está a tornar-se cada vez mais popular tendo crescido cerca de 350% em 2018 [5].

[2] Um *patch* é um fragmento de código que serve para corrigir problemas de segurança ou problemas relacionados com funcionalidades de *software*.

vez que estas oferecem uma forma mais rápida e interativa de comunicação.

## 1.3 Contribuições

A contribuição deste projeto foi o sistema RoboCISO. Para atingir os objetivos enunciados o RoboCISO foi desenhado e desenvolvido em estreita colaboração com o CISO da Altice Portugal de forma a responder às necessidades por ele identificadas. O sistema implementa casos de uso nas diferentes atividades do CISO que podem ser dividas em três vertentes: **Urgência**, **Atenção** e **Rotina**. Em seguida estão descritos e distribuídos pelas três vertentes os vetores abordados neste projeto:

1. **Urgência** - Nesta vertente estão assentes os processos de alarmística, relativos ao que está a acontecer no momento, como por exemplo um ciberataque em curso. Os possíveis problemas que fazem parte da vertente de urgência não devem depender de terceiros para chegarem ao conhecimento do CISO, devendo existir um processo automático que o alerte atempadamente.

   Nesta vertente são abordados dois vetores. O primeiro é relativo aos ataques DoS/DDoS. No início do projeto o CISO era alertado, via SMS, acerca dos ataques de DoS/DDoS em curso mesmo aqueles sem uma expressão significativa, causando um excesso de alertas. O objetivo foi desenvolver um processo que envie alertas apenas quando estes apresentam uma dimensão que represente realmente uma urgência e, preferencialmente, que o meio de comunicação fosse mais eficiente que o SMS, dado que este contém muita informação e para compreender qual a gravidade do ataque é preciso analisá-lo em toda a sua extensão.

   O segundo vetor abordado é relativo ao estado das componentes e plataformas críticas para o funcionamento da organização. O CISO deve ser informado em tempo útil se uma das suas plataformas/componentes críticas deixar de funcionar. O foco deste caso de uso foi nas seguintes plataformas:

   (a) **High Performance Infrastucture for Data Research** (HIDRA) é uma base de dados não relacional de alto desempenho, disponibilidade e escalabilidade utilizada para análise de segurança, baseada principalmente em *machine learning* e visualização, para apoiar a investigação de fraudes e resposta a incidentes de segurança, desenvolvido pela DCY [6]. Sendo uma das principais bases de dados utilizadas, se não estiver em funcionamento é de extrema importância que o CISO seja alertado pois não estarão a ser registados eventos e toda a informação ali presente poderá ficar indisponível.

   (b) **RoboCISO**, uma vez que este é a nova fonte de alertas, o CISO será apenas notificado quando os *thresholds* são ultrapassados, ou seja, podem existir espaços de tempo onde não são gerados alertas/notificações contudo, esta fonte estará em funcionamento. A ausência de alertas pode ser boa, significando que não existem problemas ou pode ser má, significando que o sistema RoboCISO não está a detetar situações de alerta ou estas situações apesar de detetadas não estão a ser reportadas, daí a importância de verificar o seu estado.

   (c) **Conjunto de plataformas/componentes que enviam eventos para o AlienVault** [3] que é o

[3] https://otx.alienvault.com/

*Security Information and Event Management* (SIEM), onde são registados os eventos produzidos por diversas plataformas críticas. Se uma ou mais dessas plataformas parar o envio de eventos para o SIEM durante um certo período de tempo, o CISO deverá ser informado da possível existência de problemas dessa(s) plataforma(s)/componente(s).

2. **Atenção** - Nesta vertente estão inseridos vetores críticos, que de momento ainda não representam uma urgência, mas se continuarem negligenciados poderão vir a sê-lo. Foram abordados dois vetores distintos:

   (a) **Cumprimento da instalação dos *Patches* da Microsoft** - Foi definido e implementado um processo que informe o CISO em relação ao estado das atualizações das máquinas com sistema operativo Windows. No início do projeto não existia nenhum processo que informasse o CISO sobre esta matéria e, dado que a Altice Portugal já sofreu ciberataques bem sucedidos no passado, é de extrema importância reduzir a probabilidade de que tal aconteça novamente. Através da implementação de um processo que notifique com regularidade o estado das atualizações, o CISO poderá tomar medidas, caso necessário e de forma atempada, evitando que o atraso das atualizações se torne uma urgência. O CISO pretende ser avisado, numa notificação matinal, se o processo crítico de *patching* está atrasado face ao estipulado.

   (b) **Cumprimento do SLA dos *tickets* do *Request Tracker Incident Response* (RTIR)** - O SLA é um compromisso entre um prestador de serviços e um cliente. Os SLAs podem ser definidos em diferentes aspetos do serviço como a qualidade, disponibilidade, responsabilidades, entre outros [7]. Neste projeto foi abordado o SLA de tempo máximo de resolução de incidentes internos, sendo a *Computer Security Incident Response Team* (CSIRT) a equipa responsável pelo cumprimento destes SLAs.

3. **Rotina** - Nesta vertente estão inseridas as tarefas mais frequentes e que têm de ser realizadas diariamente. Neste projeto foi abordada apenas uma dessas tarefas:

   (a) **Leitura dos *heatmaps* da BitSight** - A DCY contratou a Bitsight para monitorizar a rede informática exposta de todo o Grupo Altice Portugal. A DCY consegue assim observar as alterações dos *ratings* Bitsight das diversas empresas do grupo bem como de empresas externas com as quais colaboram. Neste caso o foco foi apenas nos *ratings* internos, ou seja, empresas do grupo Altice Portugal, gerando um alerta se existir uma alteração acentuada ao longo do dia (seja esta alteração positiva ou negativa).

## 1.4 Estrutura do documento

Este documento seguirá a estrutura abaixo descrita:

- Capítulo 2 - **Contexto e trabalho relacionado** - O propósito deste capítulo é fazer uma breve introdução aos principais temas relacionados com o âmbito deste projeto. O primeiro tema é a *alert fatigue*, algo que queremos evitar que aconteça com o RoboCISO. O segundo tema abordado é o *patching* na Microsoft, dado que foram implementados cinco processos sobre este tema é

relevante existir uma secção dedicada ao mesmo. Na Secção 2.3 é feita uma breve introdução ao conceito de *Situational Awareness* e na Secção 2.4 uma breve descrição dos pilares da gestão da cibersegurança na Altice Portugal.

- Capítulo 3 - **Arquitetura RoboCISO** - Neste capítulo é explicada em detalhe a arquitetura RoboCISO bem como a arquitetura de cada um dos processos desenvolvidos.
- Capítulo 4 - **Implementação RoboCISO** - Neste capítulo é explicada em detalhe a implementação dos sistemas descritos no Capítulo 3.
- Capítulo 5 - **Resultados e Avaliação** - Neste capítulo são apresentados os resultados da avaliação aos sistemas desenvolvidos.
- Capítulo 6 - **Conclusão** - Este capítulo contém um sumário do que foi realizado neste projeto e apresenta algumas conclusões que podem ser retiradas do desenvolvimento dos sistemas RoboCISO, contendo também uma secção dedicada ao possível trabalho futuro.

# Capítulo 2

# Contexto e trabalho relacionado

O sistema desenvolvido neste projeto tem como principal objetivo manter o CISO informado sobre diferentes vetores de risco, sendo por isso a sua principal função a geração de notificações e alertas para cinco casos de uso diferentes, onde são verificadas várias plataformas e componentes críticas ao funcionamento de toda a infraestrutura. Um dos principais desafios de um sistema de alerta é evitar a *alert fatigue*, ou seja, o excesso de alertas. É igualmente importante garantir que a não existência de alertas é de facto algo positivo e que não significa que o próprio sistema de alerta está com problemas e portanto não está a desempenhar as suas funções como esperado. Para cada processo existem condições distintas para a geração dos respetivos alertas. Dependendo do caso de uso a tratar, existem *thresholds* diferentes, sendo que estes não podem ser demasiado baixos, pois corremos o risco de gerar demasiados alertas, nem demasiado altos pois arriscamo-nos a deixar passar situações cujo alerta seria relevante.

Desta forma, na Secção 2.1 deste capítulo é feita uma introdução ao tema da *alert fatigue*, utilizando um estudo feito pela *CRITICALSTART* [1], em 2019. Na Secção 2.2 é feita uma breve descrição de como é feito o processo de lançamento de *patches* da Microsoft, uma vez que dos nove processos criados, cinco estão relacionados com este tema. Na Secção 2.3 é feita uma breve introdução ao conceito de *Situation Awareness*, um tema que faz parte dos objetivos do RoboCISO uma vez que o intuito deste é dar uma visão situacional ao CISO do estado dos seus vetores mais críticos. A Secção 2.4 é dedicada à gestão da cibersegurança da Altice Portugal onde é feita uma breve descrição dos cinco pilares que a compõem.

## 2.1 *Alert Fatigue*

Em 2019, a CRITICALSTART conduziu um estudo que contou com a participação de mais de 50 profissionais de *Security Operations Center* (SOC) de diversas organizações, *Managed Security Services Providers* (MSSP) e também fornecedores de serviços de *Managed Detection and Response* (MDR) de forma avaliar o estado da resposta a incidentes dentro dos SOC [8]. Este estudo é a continuação de um estudo semelhante realizado em 2018, e concluiu-se que os analistas SOC continuam a ter de enfrentar um número esmagador de alertas todos os dias e que estão a demorar mais tempo a investigá-los. Como resultado, muitos desses analistas acreditam que a sua função primária no trabalho seja "reduzir o tempo necessário para investigar os alertas". De forma a lidar com o aumento de alertas, as organizações tendem a contratar mais analistas ou a incentivar os existentes a ignorarem certos tipos de alertas e a desligarem

[1] Empresa que fornece serviços de deteção e resposta a incidentes.

certas opções nas plataformas para não serem gerados tantos alertas [8].

Grande parte dos líderes na área de cibersegurança recebem mais de 10 alertas por dia, o que está a aumentar o risco da existência da *alert fatigue*. Este estudo apurou que 70% dos analistas viu um aumento do número de alertas no último ano (2019). Em 2018 apenas 45% tinham reportado investigar mais do que 10 alertas por dia. Na Figura 2.1 estão representados os resultados obtidos relativamente à pergunta "Quantos incidentes/alertas investiga por dia em média?", neste estudo de 2019.

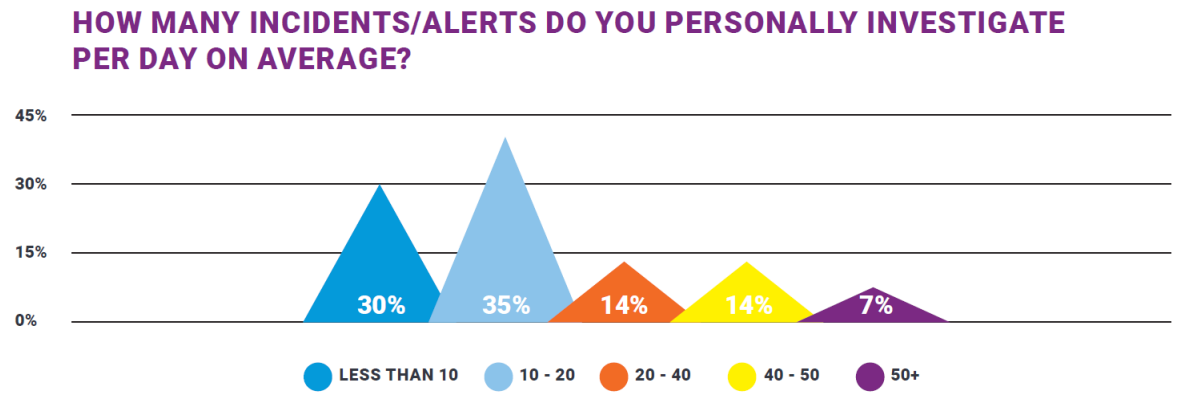



Figura 2.1: Número de incidentes investigados diariamente pelos analistas, extraído de [8]

Cerca de 78% dos analistas revelam que demoram 10 minutos ou mais para investigar cada alerta, como ilustrado na Figura 2.2. Apenas 22% dos participantes disse que demora menos de 10 minutos o que representa uma descida de 36% em comparação com o estudo realizado no ano de 2018. Em média os investigadores descobriram que quem responde aos alertas demora cerca de 2 horas e 30 minutos a 5 horas por dia investigando os mesmos.

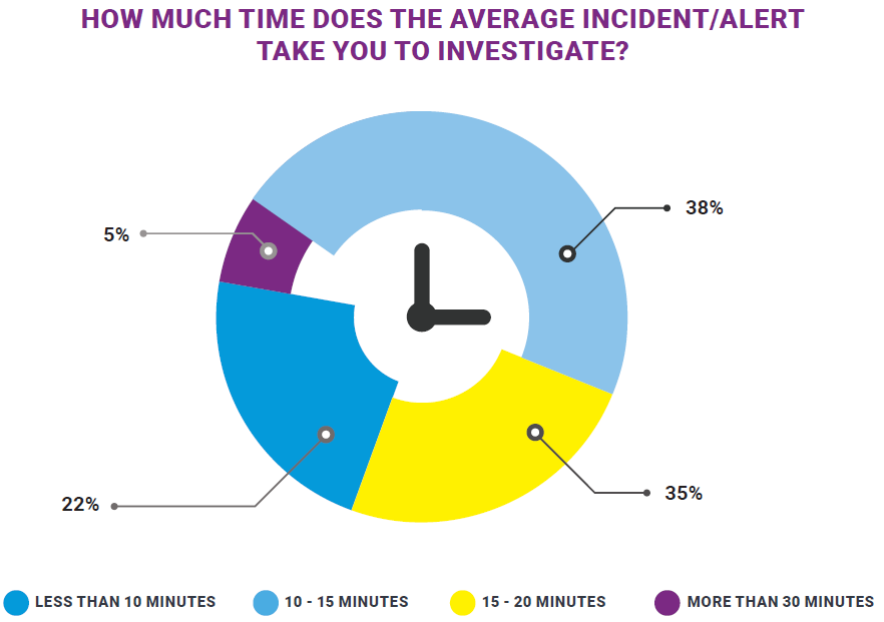



Figura 2.2: Intervalo de tempo necessário para investigar um alerta, extraído de [8]

Ainda no estudo em [8], os investigadores afirmam que à medida que a infraestrutura de IT se tornou

mais complexa de proteger contra as ameaças que surgem constantemente e a grande velocidade, combinado com um mercado de trabalho ainda muito restrito no que toca a profissionais de cibersegurança, as organizações recorrem cada vez mais a fornecedores de serviços de segurança para complementar e estender as suas competências em cibersegurança e gestão de risco. Esta abordagem consiste simplesmente em passar a responsabilidade de analisar esse elevado número de alertas para uma organização externa que fornece esse serviço. Agravando o problema está depois o aumento dos falsos positivos, que correspondem, segundo o estudo, a mais de metade dos alertas diários. Este número continua alto pois os sistemas de resposta a incidentes e SIEMs continuam a gerar alertas para atividades comuns e seguras apesar de terem como objetivo detetar eventos suspeitos. Para além disso, estas ferramentas têm como um dos seus objetivos agregar e coordenar a informação recolhida para analisar e processar os alertas, mas não abordam os desafios impostos pelo aumento de alertas ou aumento de falsos positivos. Os resultados relativamente à pergunta "Tipicamente, qual a percentagem de alertas que investiga que são falsos positivos?", para o ano de 2019 estão ilustrados na Figura 2.3.

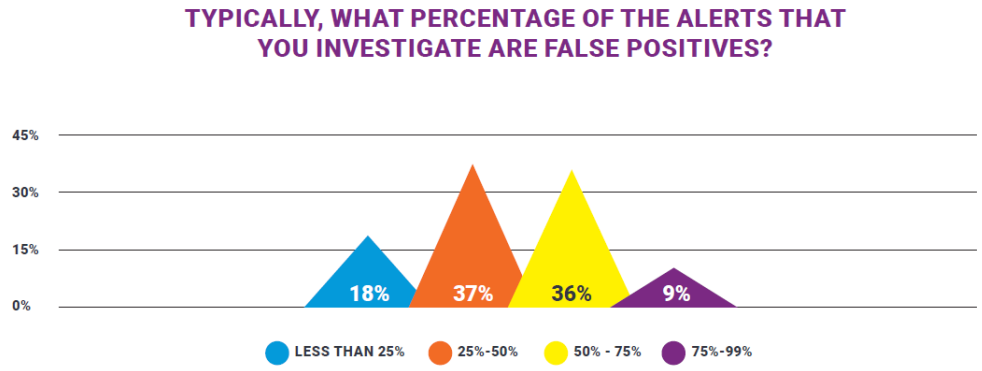



Figura 2.3: Percentagem de falsos positivos analisados, extraído de [8]

Tal como ilustrado na Figura 2.4, de forma a tentar minimizar a *Alert Fatigue*, 57% dos analistas disse que desativar certas opções nos sistemas e definir limites para os valores dos diferentes parâmetros pode reduzir o seu volume. Ignorar certas categorias de alerta faz parte da estratégia de 39% e desativar as opções que produzem mais alertas é feito por 38% dos analistas inquiridos. Foram também contratados mais analistas SOC e como resultado, o quíntuplo dos inquiridos em 2018, agora em 2019 respondeu que investigar os alertas é a sua principal função no trabalho.

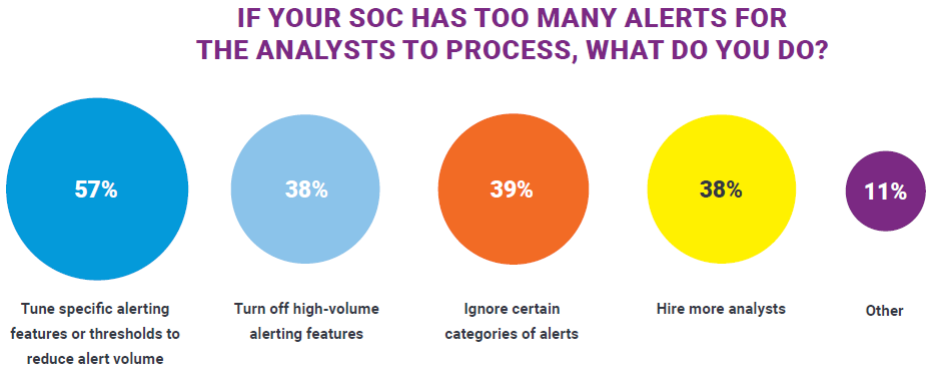



Figura 2.4: Ações tomadas em caso de excesso de alertas, extraído de [8]

O problema de excesso de alertas vai para além do nível de segurança da organização e tempo de investigação dos analistas, afetando os modelos de negócio, impactando a equipa e os processos. As próprias organizações de MSSP e MDR continuam a ter dificuldades em perceber qual a melhor forma de gerir este excesso de alertas, o que resulta na contratação de mais analistas ou na alteração de definições das aplicações de forma a que estas produzam menos alertas. Os investigadores observaram também que a *alert fatigue* e o tempo despendido na investigação dos alertas aumentou a rotação das pessoas nessas organizações. Como representado na Figura 2.5, 80% dos líderes disse que experienciou uma rotação de analistas em 10%, e 45% disse que experienciou uma rotatividade de 10% a 25% [8].

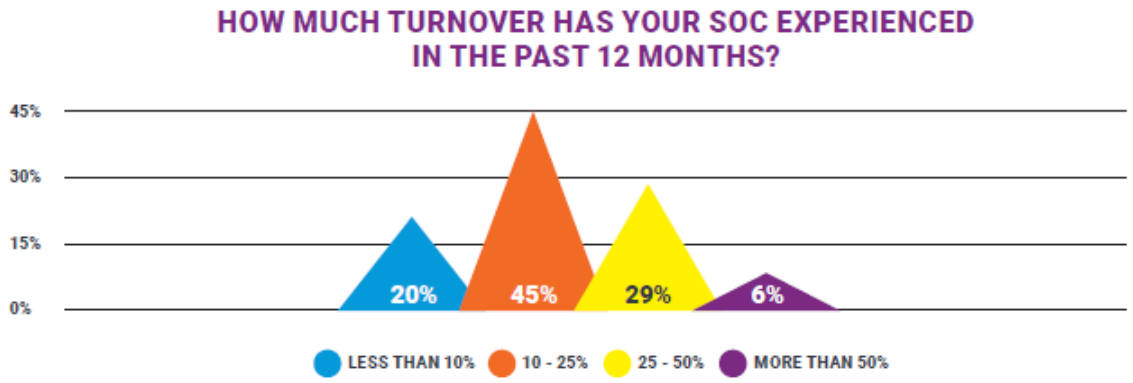



Figura 2.5: Rotação nas equipas de analistas,extraído de [8]

Apesar destes estudo ser focado nos analistas SOC este problema estende-se pelos diferentes cargos na área de cibersegurança, nomeadamente pelo CISO. Desta forma, um dos grandes objetivos deste projeto é evitar esta *alert fatigue* neste caso não focada nos analistas SOC, mas sim no CISO, sendo que como trabalho futuro poderá estender-se pelas diferentes áreas da organização.

## 2.2 Gestão de *Patching*

Um *patch* trata-se de uma atualização ao código de um dado *software* para corrigir erros descobertos após o lançamento do produto. Um *patch* de segurança, tal como os restantes *patches* faz atualizações no código do software, no entanto, estes corrigem vulnerabilidades no *software* que criminosos informáticos podem utilizar para obter acesso ao dispositivo e aos dados dos utilizadores [9].

### 2.2.1 *Timing*, Priorização e Testes

O *Timing*, a Priorização e os Testes são os grandes três problemas da gestão de *patching*. Idealmente uma organização deveria instalar os novos *patches* imediatamente após serem lançados, de forma a minimizar o tempo em que os sistemas estão vulneráveis após a divulgação pública das vulnerabilidades. No entanto, na realidade este tipo de abordagem não é possível pois as organizações dispõem de recursos limitados, o que torna necessária a existência de um sistema de priorização de *patches*. Para tornar esta gestão ainda mais complexa é bastante arriscado instalar *patches* sem que estes sejam testados, pois podem causar interrupções operacionais que acabam por se tornar mais danosas do que o impacto causado pela não instalação do *patch* tão atempadamente. Uma vez que fazer estes testes para além de tempo

também consome os recursos limitados da organização, torna-se fundamental a existência da priorização anteriormente referida. Desta forma existe sempre um conflito entre estes três fatores [10].

Os próprios fornecedores de *software* começaram a diminuir este conflito melhorando a forma como estes são distribuídos. Atualmente os *patches* são lançados em grupo em vez de serem lançados individualmente. Esta abordagem permite que a organização execute o seu plano de testes e os instale de uma só vez, ao invés de terem de o fazer várias vezes por semana, tornando o processo mais eficiente e reduzindo a necessidade de priorizar *patches* individuais, priorizando por grupo de *patches*. Ainda assim existem alguns pontos negativos nesta abordagem, como por exemplo o tempo desde que uma vulnerabilidade é descoberta até que o *patch* fique disponível publicamente aumenta. Se um atacante descobre essa mesma vulnerabilidade antes do *patch* ser lançado, aumenta a janela de oportunidade para explorá-la [10].

Existem ainda mais problemas a considerar em relação ao *timing*. O lançamento de um *patch* pode dar aos atacantes a informação que estes necessitam para poder explorar uma vulnerabilidade, o que significa que o *patch* recentemente lançado deverá ser instalado de imediato para evitar que o sistema seja comprometido. No entanto, se uma vulnerabilidade ainda não foi explorada, as organizações devem pesar cuidadosamente o ciber risco de não instalar o *patch* com o ciber risco de instalar sem fazer testes primeiro. Outro aspeto é o tempo que este *patch* demora a ter efeito, uma vez que pode ser necessário forçar a implementação de mudanças, o que pode requerer a reinicialização do serviço ou aplicação ou mesmo o *reboot* do sistema operativo. Em última análise, o que importa não é quando o *patch* foi instalado, mas sim quando o efeito deste entra em vigor na máquina [10].

Priorizar que *patch* instalar e quando instalar relaciona-se com o *timing*, mas existem outros aspetos a ter em consideração, como a importância relativa dos sistemas vulneráveis (se são servidores ou *endpoints*) e a severidade relativa de cada vulnerabilidade, por exemplo o *Common Vulnerability Scoring System* (CVSS)[2]. Outro aspeto a ter em consideração são as dependências que os *patches* podem ter entre si, instalar um dado *patch* pode requerer instalar outros primeiro, e em alguns casos pode requerer a reinicialização da aplicação ou *reboot* do *host* múltiplas vezes para que os *patches* possam ser instalados em sequência [10].

### 2.2.2 *Patching* na Microsoft

Sendo o *patching* da Microsoft o foco da maioria dos processos desenvolvidos neste projeto é relevante fazer um pequeno resumo de como se processa esta gestão do *patching*. John Wilcox, um designer da empresa publicou em 2018 um guia sobre o *patching* da Microsoft onde nomeou os seus 3 princípios [12]:

- **Serem simples e previsíveis**, de forma a que os gestores tenham a possibilidade de planear as atualizações uma vez que a cadência é regular. Deste modo, não é necessário parar tarefas a meio para instalar os novos *patches* uma vez que a sua instalação é planeada com antecedência;
- **Serem ágeis**. Atualmente no âmbito da cibersegurança, é essencial responder a ameaças rapidamente quando necessário e por isso as atualizações são fornecidas sempre que necessário sem

[1] Os *endpoints* correspondem aos *laptops* e *desktops*.

[2] O CVSS atribui a cada vulnerabilidade uma pontuação numérica que reflete a sua severidade. A pontuação numérica pode então ser traduzida numa representação qualitativa (tal como baixa, média, alta e crítica) para ajudar as organizações a avaliar e priorizar adequadamente os seus processos de gestão de vulnerabilidades [11].

comprometer a sua qualidade e compatibilidade;

- **Serem transparentes**. Para simplificar o processo de instalação de *patches* do Windows em organizações, estas devem ter o máximo de informação possível possibilitando a preparação atempada das atualizações. Esta informação inclui as notas, sistemas de *feedback* e assistência.

*Patch Tuesday* é um termo não oficial usado para referenciar o dia em que a Microsoft lança os *patches* para os seus produtos de *software*, e teve início a Outubro de 2003. Antes da existência da *Patch Tuesday*, que ocorre na segunda e, por vezes, quarta semana do mês, era seguido um modelo de "*ship when ready*", ou seja, quando o *patch* estava pronto era imediatamente lançado, e por vezes, aconteciam vários lançamentos no mesmo dia. O sistema da *Patch Tuesday* acumula *patches* ao longo do mês e estes são lançados em conjunto permitindo que os administradores de sistemas se possam preparar para as atualizações. Terça-feira às 10h00 da manhã no horário do pacífico, foram escolhidos como dia e hora ótimos, de forma a maximizar o tempo disponível entre o próximo fim-de-semana para corrigir quaisquer problemas que possam surgir com a atualização e deixando segunda-feira para resolver problemas inesperados que podem surgir durante o fim-de-semana. A Microsoft tem também durante o resto da semana em atenção o *feedback* dado pelos utilizadores e organizações de forma a preparem as correções de imediato se necessário.

Criando um padrão no que toca ao lançamento dos *patches*, a Microsoft esperava que no futuro as pessoas, e principalmente as empresas, não tivessem de adiar a instalação dos *patches* devido aos testes. No ambiente atual, isto ainda é mais problemático. Os ataques *Zero-day*[3] cresceram exponencialmente na última década, tal como a rapidez e sofisticação dos mesmos. Os atacantes usam também a data da *Patch Tuesday* contra ela mesma estabelecendo um padrão de *Patch Tuesday/Wednesday Exploit*.

A política da *Patch Tuesday* é adequada quando uma vulnerabilidade não é amplamente conhecida ou extremamente obscura, mas nem sempre é esse o caso. Existiram casos onde a informação de uma vulnerabilidade se tornou pública ou *worms*[4] que começaram a circular antes da próxima *Patch Tuesday*. Em casos críticos a Microsoft utiliza a opção "*out of band*" que consiste em lançar de imediato os *patches* permitindo a sua instalação antes da próxima *Patch Tuesday* [13].

Na Altice Portugal a política de instalação destes *patches* foi alterada após o ataque do *Wanna Cry* como referido na Secção 1.1. Todos os *patches* classificados como *Critical* pela Microsoft são lançados na segunda terça-feira de cada mês. Na quarta-feira essas atualizações são carregadas no Windows Server Update Services (WSUS)[5], que são disponibilizados em Portugal de acordo com o fuso horário. As atualizações são enviadas para uma amostra de 500 *endpoints* representativos do parque pelas 12:00h de quarta-feira com quatro horas de *deadline* para ser feito o *restart* da máquina para o *patch* entrar em vigor. Nos quatro dias seguintes, quinta a domingo, é feita a avaliação do impacto desta instalação. Por fim, ao sexto dia as atualizações são lançadas para todo o parque a partir das 7:00h da manhã. Os *endpoints* são forçados a fazer um *restart* no máximo 7 dias depois da distribuição dos *patches* de forma a garantir que estes entram em vigor na máquina. Ao fim de duas semanas todos os *patches* críticos teriam de estar instalados [14].

[3] Ataque desconhecido pelo fabricante/fornecedor do serviço e pela comunidade de cibersegurança.

[4] *Malware* auto-suficiente, que se auto-replica através de ligações à Internet.

[5] Serviço desenvolvido pela Microsoft que permite aos administradores gerir a distribuição das actualizações de produtos Microsoft para computadores em ambiente empresarial.

#### *Security Update Severity Rating System* da Microsoft

Ataques que têm impacto negativo no sistema dos utilizadores com sistema operativo Windows raramente resultam da exploração de vulnerabilidades desconhecidas. Pelo contrário, os atacantes exploram as vulnerabilidades para as quais foram distribuídos *patches* tendo como alvo máquinas onde estes não foram aplicados. Nem todas as vulnerabilidades são igualmente severas e para ajudar os utilizadores a perceberem o risco associado a cada uma, foi criado o *Security Update Severity Rating System* [15]. Este sistema classifica cada vulnerabilidade de acordo com o pior cenário (teoricamente) possível onde aquela vulnerabilidade é explorada.

| Rating | Descrição |
|---|---|
| *Critical* | Uma vulnerabilidade cuja exploração pode permitir a execução de código sem a interação do utilizador. É recomendada a instalação imediata destas atualizações. |
| *Important* | Uma vulnerabilidade cuja exploração pode comprometer a confidencialidade, integridade ou disponibilidade dos dados do utilizador ou a integridade e disponibilidade de recursos. A sua instalação é recomendada assim que possível. |
| *Moderate* | O impacto desta vulnerabilidade é mitigado até certo ponto por fatores como os requisitos de autenticação. É recomendado que os clientes considerem a aplicação da atualização de segurança. |
| *Low* | O impacto da vulnerabilidade é atenuado de forma abrangente pelas características da componente afectada. É recomendado que os clientes avaliem se devem aplicar a actualização de segurança aos sistemas afetados. |

Tabela 2.1: *Ratings* atribuídos pela Microsoft às vulnerabilidades.

## 2.3 *Situational Awareness* (SA)

Os líderes de uma organização ou direção, como o CISO, precisam de ter uma noção global significativa do que se passa na sua infraestrutura de forma a protegê-la salvaguardando os seus dados sensíveis e sustentando as suas operações fundamentais. A crescente dependência do "ciberespaço" tem aumentado, ao longo dos anos, a necessidade de estar sempre a par do que se passa, compreendendo o ambiente, prevendo e respondendo com precisão a potenciais problemas que possam ocorrer. Os sistemas e redes que operam no "ciberespaço" têm vulnerabilidades que apresentam riscos significativos para as organizações. Ao antecipar o que poderá acontecer, os líderes poderão tomar medidas em tempo útil para proteger a sua infraestrutura [16]. Desta forma o conceito de *Situational Awareness* (SA) é extremamente importante no âmbito da cibersegurança.

A SA é definida em [17] como: "*Within a volume of time and space, the perception of an enterprise's security posture and its threat environment; the comprehension/meaning of both taken together (risk); and the projection of their status into the near future*". A SA auxilia os responsáveis, sejam estes de toda uma organização ou de um departamento, fornecendo a informação necessária para que estes possam

tomar boas decisões no decurso da sua função.

Até a organização mais pequena possui ativos que deve proteger das ciberameaças, dado que são esses ativos que permitem à organização conduzir as suas atividades. É necessário priorizar os ativos de acordo com a sua criticidade para as funções da organização, de forma a protegê-los eficazmente. Nem a priorização nem a proteção podem ocorrer sem primeiro compreender o que se está a proteger, porquê, do quê e de que forma os ativos já estão ou não protegidos. Para existir uma boa proteção dos ativos tem de existir uma boa governança e boas políticas [18], tornando a SA vital para a tomada de decisões por parte da liderança.

A SA não é um conceito novo tendo sido em 1995 introduzido um modelo teórico sobre a sua aplicação [19]. Endsley, a autora, criou um modelo concetual da SA baseado na aviação mas que pode facilmente ser aplicado a outros ambientes como o da cibersegurança. Em [20] é feita também a analogia com a aviação, em tempos de guerra, onde para os pilotos, os fatores de contexto (o que está a acontecer), circunstância (o que aconteceu) e consequência (o que pode acontecer) podem convergir em segundos; para os CISOs, estas situações podem desenrolar-se ao longo de dias e meses. No entanto, estes princípios são dignos de exploração por parte de CISOs e outros que lidam com a constante mudança e complexidade. A SA é representada dividida em três níveis tal como se encontra ilustrado na Figura 2.6.

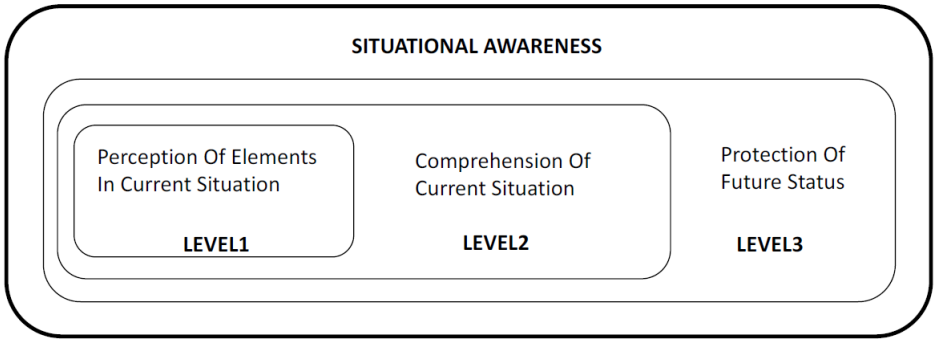



Figura 2.6: Níveis da *situational awarenss*, extraído de [21]

1. **Perceção dos elementos na situação atual** - O primeiro nível para atingir a SA é perceber o estado, os atributos e a dinâmica dos diferentes elementos de interesse no ambiente [19].

2. **Compreensão da situação atual** - O segundo nível consiste em compreender a situação atual dos diferentes elementos percecionados no nível 1. Este nível vai além do 1 pois baseado no conhecimento adquirido sobre os elementos e observando os padrões entre eles, é possível obter um conhecimento relativamente à importância desses elementos [19].

3. **Proteção do estado futuro** - A capacidade de prever as ações futuras, pelo menos a curto prazo, dos diferentes elementos do ambiente forma o terceiro e último nível para atingir a SA. Este nível é atingido através do conhecimento do estado e dinâmica dos elementos e da compreensão a situação provenientes dos dois primeiros níveis [19].

Mesmo nas organizações mais bem financiadas e mais maduras, existem lacunas de informação no conhecimento do estado atual e do que este devia ser [18]. O RoboCISO pretende preencher essas lacunas

existentes e sendo esta a primeira versão do sistema, o seu foco será apenas num pequeno conjunto de vetores não alcançando por isso uma SA global. A implementação destes primeiros casos de uso fará com que o CISO passe a estar informado com regularidade apenas sobre estes vetores, sendo que futuras versões deste sistema deverão inserir novos vetores, atingindo assim o objetivo final de uma visão global do estado da cibersegurança em toda a infraestrutura.

## 2.4 Gestão da Cibersegurança na Altice Portugal

O modelo da governança da cibersegurança da Altice Portugal, que tem o CISO no seu centro e a DCY como instrumento operacional, é composta por cinco pilares: Gonvernança Ativa; Prevenção Ativa; Deteção e Resposta Ativa; Proteção Ativa e Recuperabilidade Ativa, ilustrados na Figura 2.7.

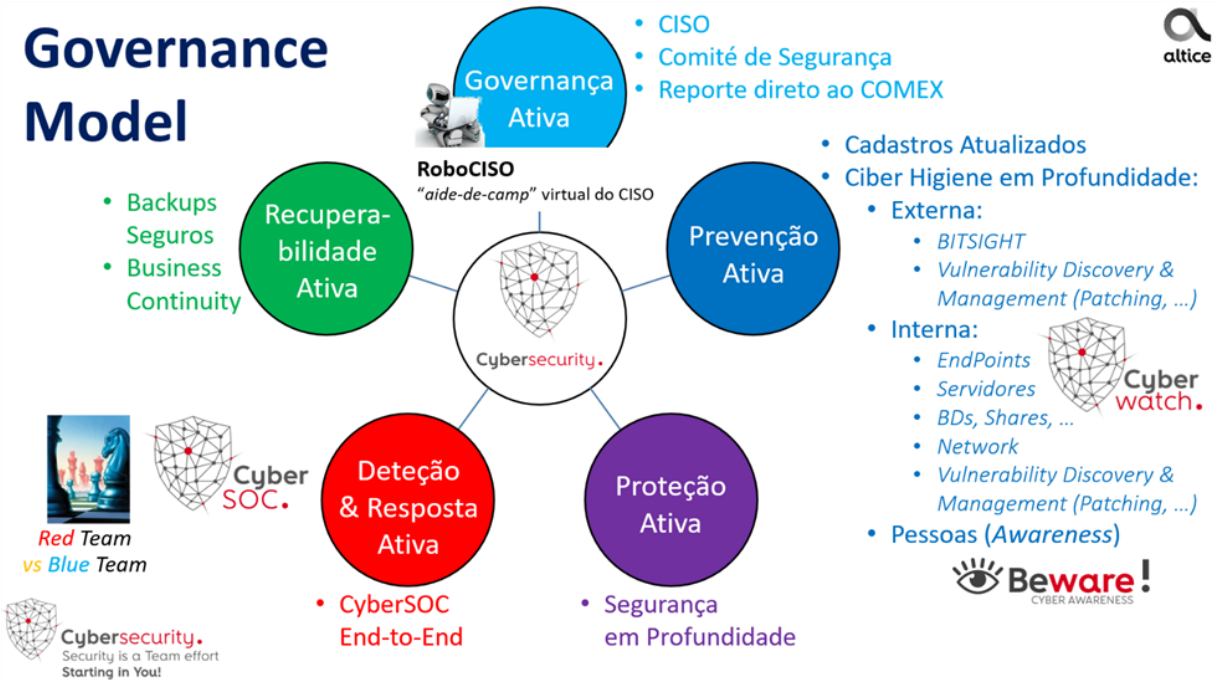



Figura 2.7: Modelo da governança da cibersegurança na Altice Portugal.

1. **Governança Ativa** - pilar central sob gestão direta do CISO e onde este pretende ter o RoboCISO como um seu assistente virtual;

2. **Prevenção Ativa** - composta por três grandes sub-pilares e respetivas equipas de suporte:

   (a) ***CyberWatch+***, orientado à melhoria contínua da "ciber higiene", externa e interna, da Altice Portugal. Neste sub-pilar está inserido o processo referente à Bitsight.

   (b) ***Beware!***, orientado à melhoria contínua do "*cyber awareness*" de todos os colaboradores internos e externos da Altice Portugal.

   (c) ***CyberWatch+ Security Analytics***, a plataforma central e *end-to-end* de "*Security Analytics*" da DCY e a principal fonte de informação do CISO e, consequentemente, do RoboCISO. Todos estes pilares alimentam, em tempo-útil, esta plataforma com as suas métricas. Com

base nestas métricas os seus indicadores são automaticamente calculados. Ambas as iniciativas anteriores alimentam diariamente esta plataforma com as métricas relevantes da sua responsabilidade. É neste pilar que o processo dedicado ao *patching* está inserido.

3. **Deteção e Resposta Ativa** (e Rápida) a incidentes - a principal componente deste pilar é o SOC da DCY. Neste pilar estão inseridos os processos referentes ao SLA e aos alertas de ataques DoS/DDoS.

4. **Proteção Ativa** contra incidentes de cibersegurança - este pilar, que, ao contrário dos três anteriores, não é da responsabilidade direta do CISO ou da DCY, mas sim da DIT (IT) ou DEO (Rede), tem como missão garantir o funcionamento das plataformas de **proteção ativa** da Altice Portugal, como VPNs, *Proxies*, *Firewalls*, plataformas de proteção contra ataques DoS/DDoS, entre outras. Neste pilar está inserido o processo relativo à verificação do estado das plataformas críticas.

5. **Recuperabilidade Ativa** Este é um pilar crítico já que dele depende a capacidade de recuperação da Altice Portugal perante situações catastróficas. Neste sentido este é um dos pilares de maior preocupação para o CISO. Contudo, não está incluído no âmbito desta tese podendo ser um tema a abordar em futuras versões do RoboCISO.

# Capítulo 3

# Arquitetura RoboCISO

Este capítulo contém a descrição da arquitetura do RoboCISO e dos seus casos de uso, sendo apresentada a proposta de implementação de cada um. A Secção 3.1 contém uma breve descrição sobre o Blue Prism, a ferramenta de automação utilizada na implementação do projeto. A Secção 3.2 contém a descrição dos requisitos do sistema RoboCISO. A Secção 3.3 descreve a arquitetura onde os diferentes processos se inserem e como interagem entre si e com as diferentes aplicações. A Secção 3.4 é dedicada ao primeiro conjunto de processos, *RoboCISO001 - Critical Patching*, que tratam a recolha da informação dos *patches*, vulnerabilidades e sua classificação em termos de severidade e a Secção 3.5 é dedicada ao conjunto de processos responsável pela deteção e geração dos diferentes alertas e notificações, *RoboCISO002 - Alerts*.

Todos os processos estão representados por *System Sequence Diagrams*[1] (SSD), cada um acompanhado de uma descrição detalhada dos passos apresentados no mesmo. Estes processos interagem com o Chrome, Outlook, base de dados relacional SQL e base de dados não relacional Elasticsearch [2].

## 3.1 Blue Prism

O Blue Prism [3] (BP) é uma plataforma de RPA construída sobre Microsoft .NET *Framework*. É usado para automatizar processos construídos em diversos tipos de plataformas entre eles: mainframe, aplicações Windows, Java e em *Web Browsers*. O *Blue Prism* é assim um conjunto de ferramentas, bibliotecas e ambientes de execução para RPA [22].

### 3.1.1 Componentes

Um processo no BP é composto por duas partes principais: um ou mais objetos de negócio que interagem com as aplicações e um processo que contém a lógica do processo.

Para a construção dos objetos é usado o *Object Studio*, onde é tratada toda a parte relacionada com interação com aplicações. Cada objeto está associado a apenas uma aplicação, definida desde início recorrendo ao *Application Modeller*, podendo esta ser de um dos vários tipos disponíveis, ex. Windows,

[1] SSD é um diagrama de sequência que mostra, para um dado cenário de caso de uso, os eventos gerados pelos atores (neste caso, robô e aplicações) e as possíveis interações entre sistemas.

[2] https://www.elastic.co/pt/elasticsearch

[3] https://www.blueprism.com/pt/

*Web*, Java, entre outros. Dentro do *Application Modeller* os diferentes elementos da aplicação serão identificados de forma a que seja possível interagir com estes como se de um humano se tratasse. Estes elementos podem ser por exemplo caixas de texto, botões ou *links* e são identificados através da operação de *spy*, que pode ser usada em diferentes modos dependendo do tipo de aplicação usada.

Na Figura 3.1 podemos ver um *Application Modeller* onde estão identificados sete elementos que estão localizados no *Application Explorer*, à esquerda na imagem. O RoboCISO000 - *WhatsApp Interaction*, que corresponde à aplicação a ser usada, neste caso o Chrome no site do WhatsApp. O elemento *Chrome Window* corresponde à janela do *browser*, *RoboCISO Conv* corresponde à conversa do RoboCISO no WhatsApp e os restantes elementos correspondem também a componentes presentes na página do *website* que está a ser visitado. Podemos observar com algum detalhe a identificação do componente *RoboCISO Conv* que tem como função permitir ao BP saber onde está localizada a conversa do RoboCISO se forma a enviar as notificações e alertas. São identificados três atributos que ajudam a identifica-la, estes são: *Web Text*, *Web Path* e *Web Element Type*. Na Figura 3.2 vemos um fragmento da página que está ser espiada, estando identificado com um retângulo a azul o componente *RoboCISO Conv* identificado no *Application Modeller*. Após a identificação de todos os elementos necessários para completar uma dada tarefa, é possível utilizar objetos específicos para interagir com os mesmos, algo que é abordado com mais detalhe na sub-secção ao *Object Studio*.

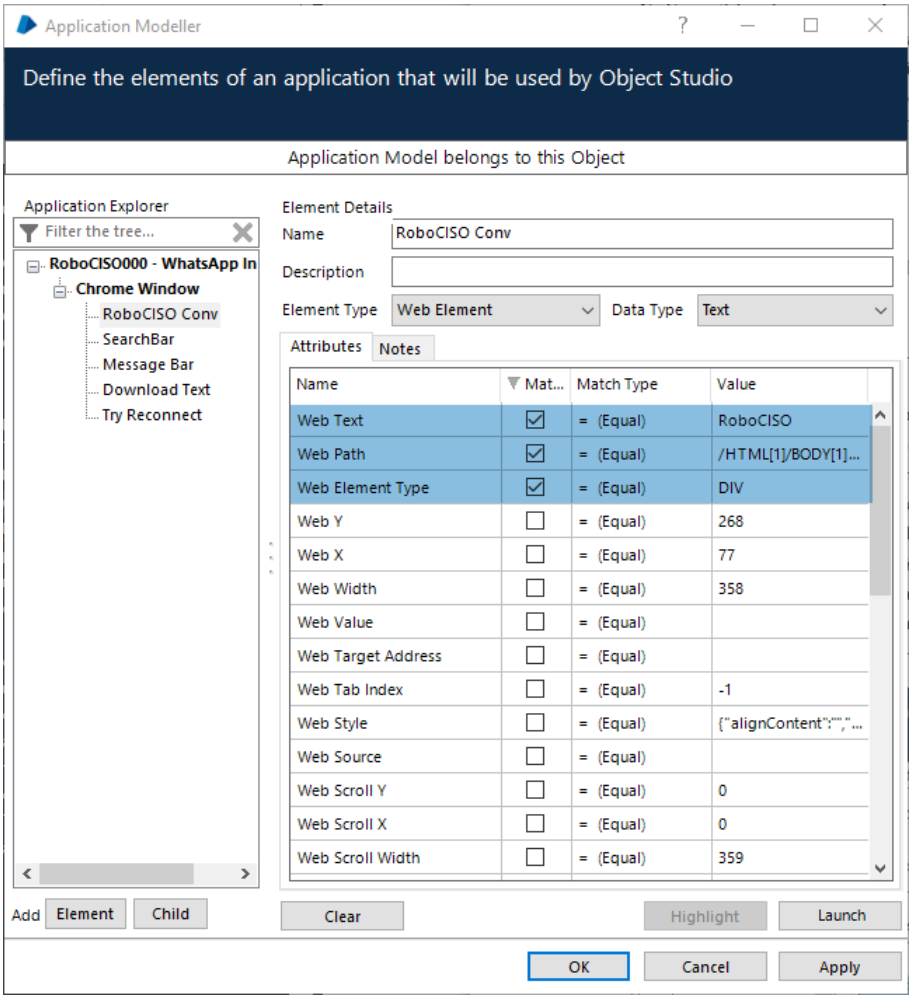



Figura 3.1: Exemplo de elementos identificados com o Application Modeller

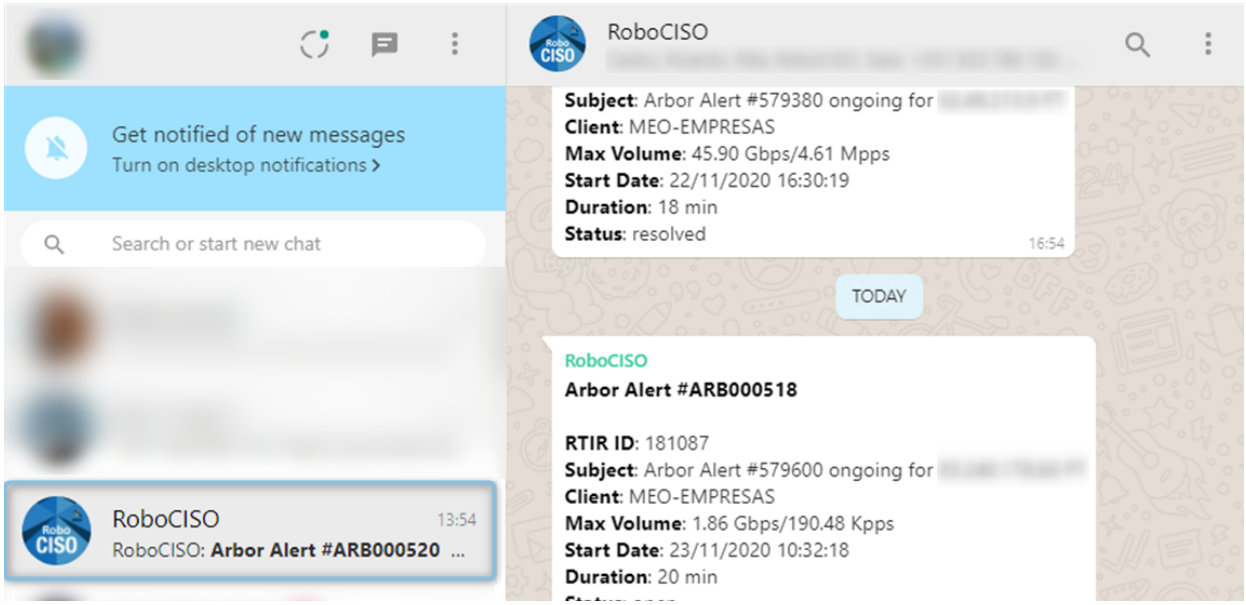



Figura 3.2: Fragmento da janela espiada pelo *Application Modeller*

Para a construção dos processos é usado o *Process Studio*, onde são definidos todos os passos lógicos do processo. Cada passo poderá ser composto por uma ação que corresponderá a um objeto de negócio. Uma vez que os objetos e o processo estejam concluídos, o robô pode ser executado. Se um processo de negócio for alterado então o processo no robô terá também de ser alterado. O mesmo acontece caso a aplicação utilizada sofra alterações, neste caso o objeto de negócio terá de ser alterado.

### *Object Studio*

Como referido anteriormente, os objetos de negócio têm como finalidade a interação com aplicações. Para alcançar este objetivo, cada objeto tem duas partes principais:

- Um *Application Model* onde é definida a aplicação a usar e os elementos da sua interface de utilizador são identificados;
- Uma ou mais páginas, onde cada uma implementa toda ou parte da operação que esse objeto de negócio executa.

Cada página de um objeto de negócio implementa uma ação. Estas ações podem depois ser invocadas por um ou mais processos, necessitando de ser publicadas (*published*), de modo a torná-las visíveis no *Process Studio*. Cada página começa sempre com a etapa *Start* e acaba numa ou mais etapas *End*, onde são também definidas as variáveis de entrada e saída, respetivamente. Entre estas duas etapas estão todos os passos necessários para completar a tarefa. Após a identificação dos elementos das aplicações, podem ser usados objetos especiais de forma a interagir com estes como se um humano se tratasse, *i.e.* escrever o nome de utilizador e palavra-passe, clicando no botão de *login* de seguida. Alguns dos objetos mais usados para conceber as ações são:

- ***Read***: faz a leitura do valor do elemento da interface de utilizador e guarda-o num *data item*;
- ***Write***: faz a escrita de um dado valor num elemento da interface de utilizador;

- ***Navigate***: abre menus, clica em botões, e realiza outras tarefas que requerem a navegação pela interface da aplicação;
- ***Wait***: pausa a execução até que uma dada condição seja satisfeita na aplicação;
- ***Code***: contém código arbitrário escrito em linguagens como C# e *Visual Basic*. Isto pode ser usado para fazer uma manipulação mais complexa sobre a informação, podendo também ser usado para aceder a bases de dados.

É também de notar que estes objetos são uma das grandes diferenças entre o *Object Studio* e o *Process Studio*, como podemos ver na Figura 3.3. O objeto *Alert* é o único objeto que não está disponível no *Object Studio* e tem como função enviar notificações a um ou mais subscritores via e-mail indicando que algo correu mal durante o processo por exemplo: enviar um alerta sempre que ocorrer uma exceção num objeto de negócio.

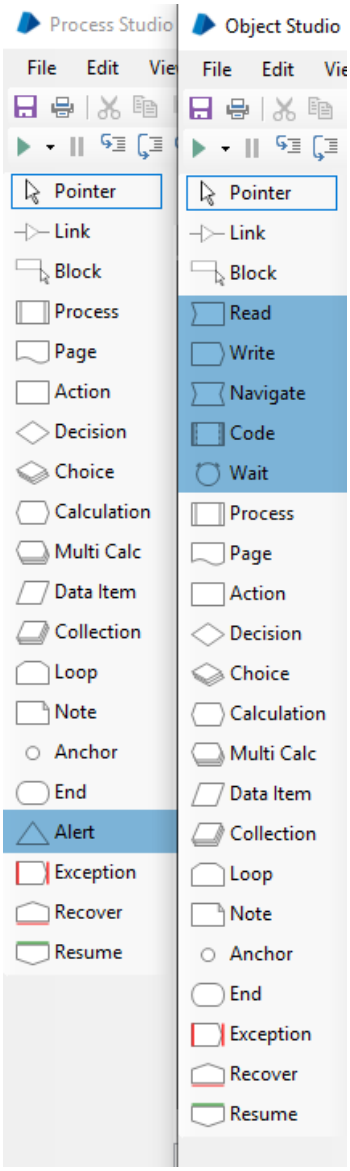



Figura 3.3: Diferença entre objetos disponíveis no *Object Studio* e no *Process Studio*

### *Process Studio*

Um processo consiste fundamentalmente numa sequência de passos a realizar pelo robô que imitam os passos que seriam realizados por um humano. Assim, tal como um humano pode interagir com várias aplicações para alcançar um dado objetivo, um robô no *Blue Prism* também o pode fazer. Um dos objetos mais utilizados no *Process Studio* é a *Action* que permite que os objetos de negócio sejam invocados. De certa forma, um objeto de negócio e um processo são semelhantes. Ambos são definidos usando uma ou mais páginas, no entanto, as páginas de um objeto de negócio podem ser invocadas em qualquer ordem, enquanto que, no caso de um processo, este começa sempre na sua *Main Page*, e as suas páginas são sempre executadas numa determinada ordem, como ilustrado na Figura 3.4.

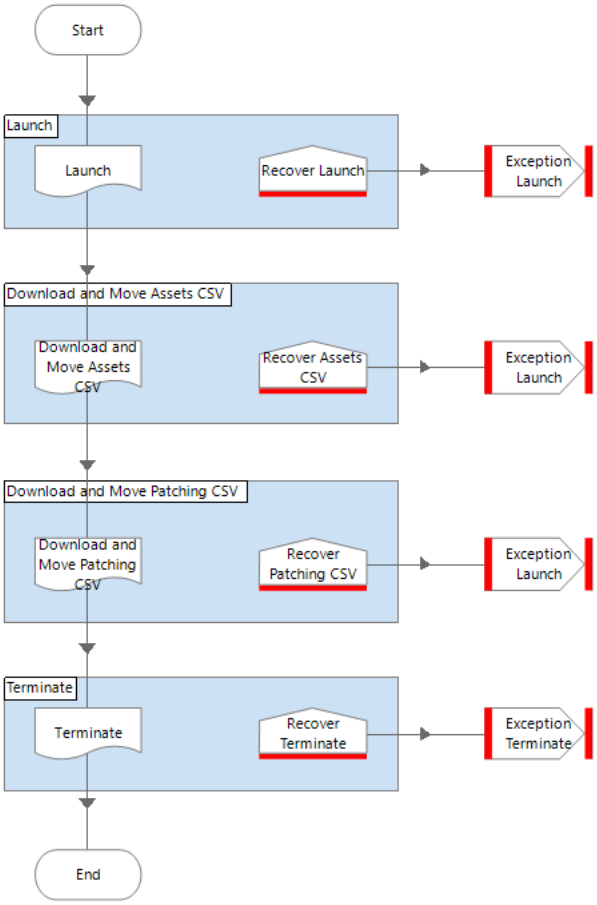



Figura 3.4: *Main Page* de um processo

### *Control Room* e *System*

Tal como nos processos manuais, os processos automatizados têm de ser controlados e geridos, para tal o *Blue Prism* dispõe de duas ferramentas: *Control Room* e e *System*.

O *Control Room* permite-nos executar tarefas como:

- Começar ou parar instâncias de robôs;
- Ver os *Session Logs* produzidos por cada robô, enquanto este está em execução ou depois de ter

terminado. Estes registos guardam informação relativa a quando é que um dado passo do processo foi executado e que informação foi utilizada;

- Criar agendas para a execução dos robôs, por exemplo: todos os dias às 09:00h e 18:00h o RoboCISO001 irá ser executado enviando um relatório com a informação recolhida ao CISO;
- Ver as filas de trabalho (*work queues*), que permitem saber exatamente até onde é que a informação foi processada e se existiram exceções no processamento da informação.

O separador *System* é focado na configuração do ambiente *Blue Prism*. É onde se podem adicionar novos utilizadores ao BP e alterar as permissões desses utilizadores. Esta é uma parte importante pois dependendo do papel do utilizador este poderá ter, ou não, acesso a determinados processos e objetos.

É no separador *System* que se criam as filas de trabalho (*queues*) e se definem variáveis globais que podem ser depois utilizadas pelos diversos objetos e processos. Estas variáveis globais podem ser credenciais para fazer *log in* em aplicações. Definindo uma variável de sistema com o tipo *password* e com um nome de utilizador ou email com o tipo *text* associado, pode-se criar uma credencial. Esta credencial é uma forma mais segura de diversos utilizadores poderem usar estes dados para acederem a aplicações sem que seja necessário saberem a palavra passe.

Toda a informação acerca dos processos, objetos, variáveis e filas de trabalho está armazenada numa base de dados própria do Blue Prism [22].

## 3.2 Requisitos do sistema

De forma a atingir os objetivos enunciados para este projeto foi definido que o sistema RoboCISO deve cumprir um conjunto de requisitos como se enuncia de seguida:

1. As notificações e alertas devem ter uma periodicidade adequada de forma a evitar a *alert fatigue*;
2. As notificações e alertas devem conter toda a informação relevante incluída e devidamente assinalada assegurando que notificações/alertas relativos ao mesmo tema são enviados numa única mensagem/*e-mail* sempre que possível;
3. Os alertas referentes a: ataques DoS/DDoS; alterações de *ratings Bitsight* ou *Health Checks* devem ser reportados em tempo útil;

Como *input*, o sistema desenvolvido deverá ter o HIDRA e as restantes bases de dados necessárias onde estão registados os eventos e informação de ativos da organização. Como meio de *output* deverá ter o WhatsApp[4]. O WhatsApp é uma plataforma de mensagens instantâneas que permite aos utilizadores trocarem mensagens escritas ou de voz, imagens, vídeos, partilhar a sua localização, partilhar documentos, entre outros. A aplicação corre em dispositivos móveis, mas permite o acesso via *web*, desde que o utilizador mantenha o telefone com acesso à Internet. Este é o segundo meio utilizado para envio das notificações e alertas do RoboCISO, pois permite uma comunicação mais rápida, direta e em tempo-real do que aquela que acontece via *e-mail*. Para além das vantagens na forma de comunicação o

[4]https://www.whatsapp.com/

WhatsApp implementa "cifra ponto-a-ponto" (*end-to-end encryption*), ou seja, as mensagens e chamadas estão protegidas para que apenas o emissor e o recetor as possam ler ou ouvir, não sendo intercetadas por ninguém, nem mesmo pelo WhatsApp [23]. Em Janeiro de 2018 foi lançado o WhastApp Business dirigido a pequenos negócios [24] e neste projeto utilizado como meio de *output* do RoboCISO. Esta versão da aplicação foi escolhida como sendo mais adequada do que a versão convencional, pois permite a criação de um catálogo. Este catálogo é usualmente utilizado para publicitar os artigos vendidos pela empresa sendo que no caso do RoboCISO foi utilizado para descrever o significado dos campos de cada alerta/notificação. Um exemplo de um dos elementos presentes no catálogo encontra-se na Figura C.1 no Apêndice C.

Para cada mensagem enviada através do WhatsApp deverá existir um *e-mail* complementar que será facilmente associável através da utilização de um identificador único. Por fim, deverá ser mantido um histórico no HIDRA de todos os alertas enviados. Esse histórico poderá depois ser utilizado para a análise e criação de estatísticas sobre as notificações e alertas enviados.

## 3.3 Arquitetura Geral

O sistema RoboCISO está inserido num sistema já existente de robôs de *software* do Blue Prism na Altice Portugal. No departamento da DCY, os processos implementados são executados em máquinas virtuais (através da VMWare[5]) sendo estas denominadas por *bots*. O RoboCISO é composto por nove processos, distribuídos por dois módulos distintos que são executados em *bots* distintos. No módulo *RoboCISO001 - Critical Patching* estão inseridos quatro processos e no módulo *RoboCISO002 - Alerts* estão inseridos cinco processos.

Na Figura 3.5 está ilustrada a arquitetura do sistema, onde é possível observar as interações entre os diferentes processos e aplicações. Todos os meses na *Patch Tuesday* é publicado no *site* da Microsoft um conjunto de *patches* e vulnerabilidades. O processo *RoboCISO001 - 01 - Get Microsoft Updates List* acede à página do *Security Update Summary* e obtém todos os novos *patches*. Após a obtenção dos *patches* a sua informação é inserida numa fila de trabalho[6] e numa tabela da base de dados, ambas dedicadas exclusivamente aos *patches*. Nesta fase os *patches* ainda não possuem um nível de severidade associado. Apesar da informação das vulnerabilidades estar presente no *site*, numa primeira fase as vulnerabilidades são obtidas através de uma *newsletter* que é enviada para o *e-mail* do RoboCISO, contendo apenas vulnerabilidades cuja severidade é *Critical*, *Important* ou *Moderate*. O processo responsável pelo processamento da *newsletter* é o *RoboCISO001 - 02 - Get CVEs from Outlook*, sendo a informação recolhida sobre as vulnerabilidades inserida numa fila de trabalho e numa tabela na BD exclusiva a estas. De seguida, o processo *RoboCISO001 - 03 - Get KBs and Severity for each CVE* itera sobre as vulnerabilidades inseridas pelo processo anterior na fila de trabalho de vulnerabilidades e, para aquelas que possuem severidade *Critical*, acede à respetiva página do *site* da Microsoft. Na página da vulnerabilidade existe uma tabela que contém para cada versão do sistema operativo Windows, o *patch* que a mitiga. Cada par de vulnerabilidade e *patch*, representado por (CVE,KB), é guardado individualmente numa fila de trabalho e numa tabela da base de dados. A cada *patch* ficará atribuído o nível de severidade *Critical*,

[5] https://www.vmware.com/

[6] Uma fila de trabalho permite preservar toda a informação processada mesmo que o processo termine em exceção, permitindo também que diferentes processos acedam à mesma informação.

que corresponde ao nível de severidade da vulnerabilidade. No processo *RoboCISO001 - 04 - Classify KBs* é atribuído um nível de severidade aos *patches* obtidos no primeiro processo (*RoboCISO001 - 01 - Get Microsoft Updates List*) fazendo o mapeamento entre esses *patches* e os obtidos no processo anterior (*RoboCISO001 - 03 - Get KBs and Severity for each CVE*).

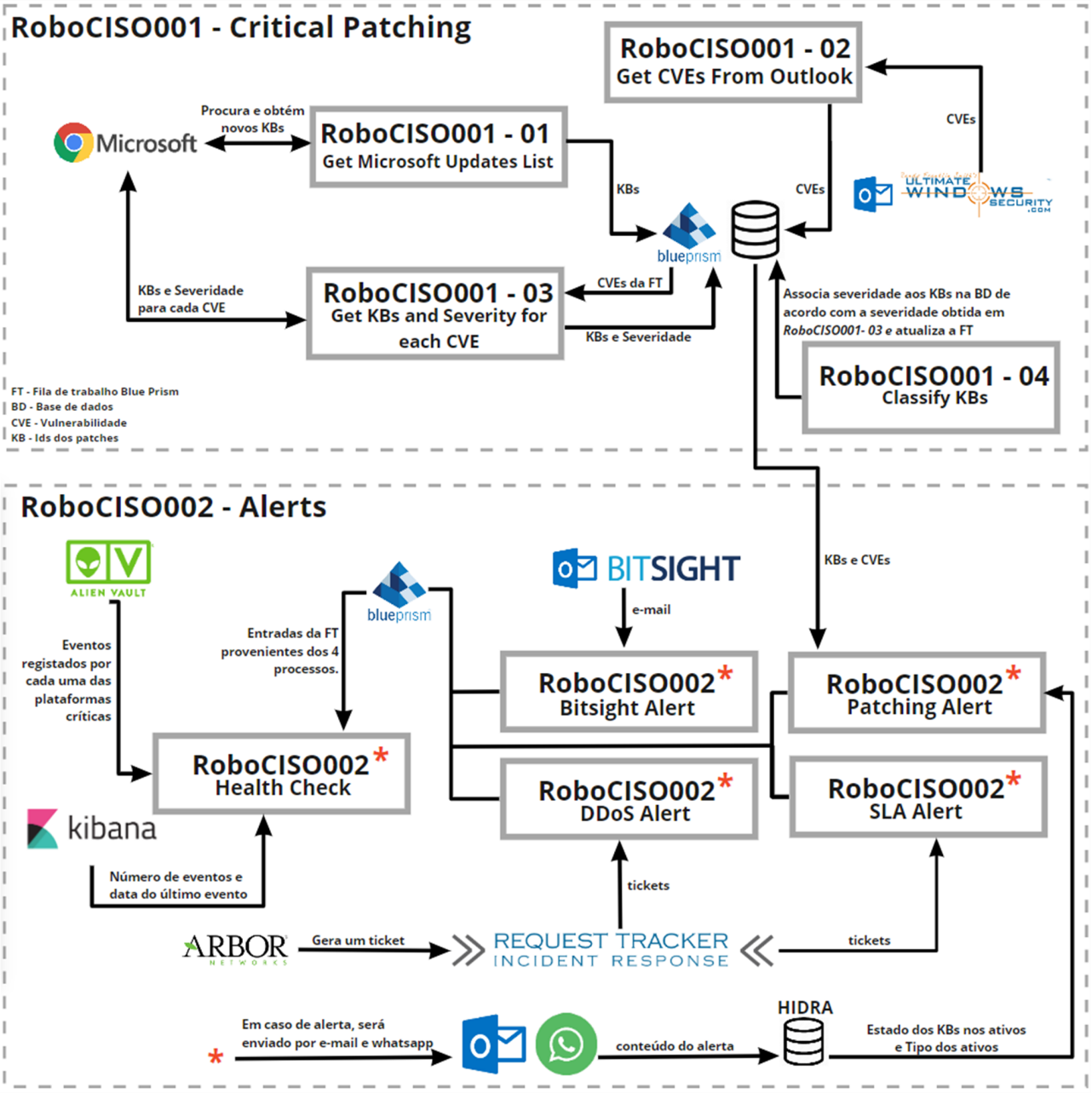



Figura 3.5: Arquitetura geral RoboCISO.

A informação obtida e processada pelo módulo *RoboCISO001 - Critical Patching* é utilizada pelo processo *RoboCISO002 - Patching Alert*, contido no módulo de *RoboCISO002 - Alerts*. Através do HIDRA é obtido o tipo dos ativos (servidor, *desktop* ou *laptop*), neste caso as máquinas com sistema operativo Windows, e o estado das atualizações nesses ativos, ou seja, quais os *patches* instalados e quais os *patches* em falta. Com ambos os conjuntos de informação é verificado se os parâmetros de alerta, detalhados na Subseção 3.5.1, são atingidos. Este processo gera notificações quer existam atualizações

em atraso ou não, informando disso mesmo. As notificações são enviadas via *e-mail* e WhatsApp acontecendo o mesmo nos restantes processos deste módulo. Sempre que é enviada uma notificação ou alerta, o seu conteúdo é inserido no HIDRA, guardando assim um histórico de todas as mensagens enviadas.

Quando existem alterações nos *ratings* da Bitsight são enviados *e-mails* para o RoboCISO. O processo *RoboCISO002 - Bitsight Alert* é responsável pelo processamento desses *e-mails*, através do Outlook, e caso os critérios de alerta sejam satisfeitos, detalhados na Subseção 3.5.4, será enviado um alerta.

Sempre que existe um incidente de cibersegurança, por exemplo uma tentativa de *phishing* reportada por um colaborador, é criado um *ticket* no RTIR. Todos os *tickets* possuem um tempo de resolução máximo associado, sendo esse o SLA aqui analisado. O processo *RoboCISO002 - SLA Exceeded* é responsável pela verificação do cumprimento desses SLAs. Se existirem situações que satisfaçam os critérios de alerta, descritos na Subseção 3.5.3, este será gerado e enviado. Um caso particular dos *tickets* do RTIR, são os *tickets* referentes a ataques de DoS/DDoS. O Arbor deteta ataques de DoS/DDoS e gera um *ticket* no RTIR. O processo *RoboCISO002 - DDoS Alert* verifica a existência de novos ataques de DoS/DDoS através dos *tickets* do RTIR. Se existir um ataque que satisfaça os critérios de alerta, descritos na Subseção 3.5.2, este será gerado e enviado.

O processo *RoboCISO002 - Health Check* verifica o estado das plataformas/componentes críticas, ou seja, se estão ou não a funcionar. Este processo interage a partir do *browser* com a interface do Alienvault (SIEM), verificando um conjunto de plataformas/componentes, descritas na Subseção 3.5.5. Para verificar o estado de cada plataforma é recolhido o número eventos detetados num dado período de tempo, por exemplo na última hora, bem como os dados do último evento registado. O processo interage também através do *browser* com o Kibana, verificando o estado do HIDRA. Para tal, recolhe o número de eventos registados nos últimos 15 minutos, bem como a data do último evento registado. Para verificar o funcionamento do próprio RoboCISO, este processo acede à fila de trabalho de alertas e verifica se os restantes processos do módulo inseriram entradas na última hora. Para estas três verificações, existem critérios de alerta diferentes que estão também descritos na Subseção 3.5.5.

## 3.4 Arquitetura RoboCISO001 - Critical Patching

No início do projeto não existia nenhum sistema que alertasse o CISO em relação ao estado das atualizações das máquinas com sistema operativo Windows, recebendo apenas via *e-mail* informação sobre os *patches* e vulnerabilidades publicados no mês corrente e o plano de instalação. Apesar de receber estes *e-mails* o CISO não sabia em que medida esses *patches* estavam a ser efetivamente instalados tornando o processo de análise e tomada de decisão mais complexo, moroso e ineficiente. O módulo *RoboCISO001 - Critical Patching* tem como objetivo obter e correlacionar toda esta informação de forma automatizada, todos os meses, e cruzá-la com a informação existente dos ativos, neste caso máquinas com sistema operativo Windows. Os processos deste módulo irão permitir que o processo *RoboCISO002 - Patching Alert* tenha toda a informação necessária para gerar notificações acerca do estado das atualizações.

### 3.4.1 Obtenção da lista de *patches* da Microsoft

Antes da existência do RoboCISO o CISO tinha acesso à informação relativa aos *patches* lançados pela Microsoft, através de um *e-mail* mensal enviado pela mesma. Estes *e-mails* apresentavam alguns proble-

mas de coerência, como o facto de num mês serem enviados em inglês e noutro em português. Ao utilizar estes *e-mails* como base para obter a informação dos *patches* a informação não ficava com uma estrutura coerente e uniformizada. Adicionalmente, o facto dos *e-mails* serem traduzidos, sendo o original em inglês, poderia implicar um atraso na sua receção de alguns dias.

No processo desenvolvido a informação dos *patches* foi recolhida a partir do *site* da Microsoft, onde os *patches* são publicados no *Security Update Summary*, cujo excerto está ilustrado na Figura 3.6, no dia da *Patch Tuesday*. Esta forma de obtenção dos *patches*, tornou a automatização do processo mais simples e o problema das diferenças de idioma dos *e-mails* foi ultrapassado. A informação recolhida a partir deste sumário foi depois inserida numa base de dados num único idioma (inglês).

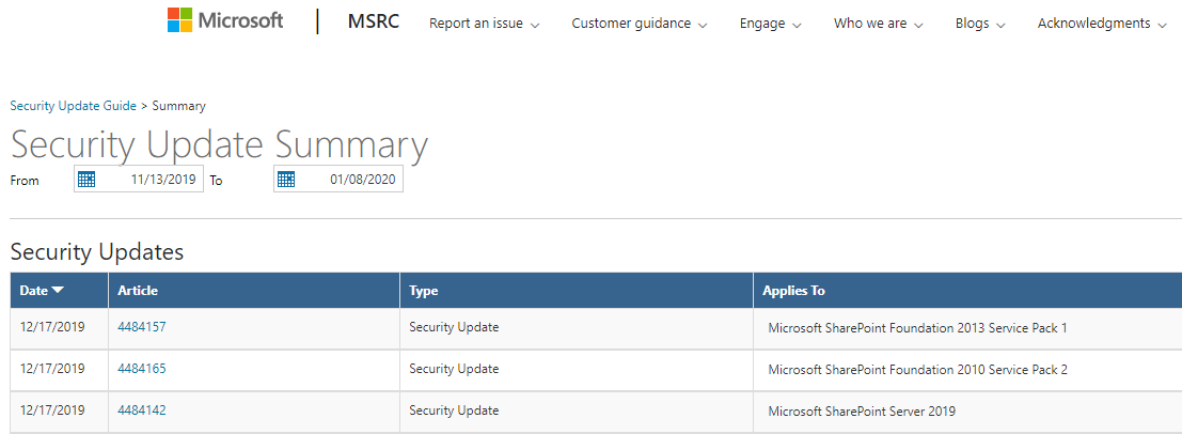



Figura 3.6: Fragmento da página *Security Update Summary* da Microsoft.

Este processo está ilustrado na Figura 3.7, sendo composto pelos seguintes passos:

1. É feito o *launch* do Chrome no *site* do *Security Update Summary*[7], *OpenBrowserAndMicrosoft()*;
2. Para aceder à base de dados, necessita de estabelecer a ligação à mesma. Para isso, estabelece uma ligação através da função *Connect(Server)*, à qual fornece o nome do servidor como parâmetro;
3. Após a ligação, é executada uma *query* do tipo *SELECT* de forma a obter a data mais recente registada. Esta data corresponde à data do último *patch* inserido na base de dados, pelo que, deverá ser a data pela qual começa a nova procura. Esta data irá definir o parâmetro *From*, que é possível observar na Figura 3.6;
4. É retornada da base de dados a data mais recente, representada por *RecentDate*;
5. O formato da *RecentDate* é passado de "yyyy-MM-dd" para a notação americana "MM/dd/yyyy", uma vez que é o formato utilizado no *site* da Microsoft e coloca-a no campo *From*, deixando o campo *To* inalterado dado que contém a data atual, *SearchPatches(RecentDate)*;
6. No *site* é apresentada uma tabela com diversas páginas, que têm de ser todas percorridas de forma a obter todos os *patches* do mês, cujos campos estão descritos na Tabela 3.1. Cada linha dessa tabela é processada individualmente e guardada numa coleção[8], *SaveInformation()*;

[7] `https://portal.msrc.microsoft.com/en-us/security-guidance/summary`

[8] Estrutura de dados do BP que pode ser interpretada como uma tabela cujos campos têm associado um tipo de dados.

7. A coleção resultante do passo anterior é iterada:

   (a) Cada item é inserido na base de dados, *Insert(Patch)*, na tabela *KB_List* previamente criada com os campos descritos na Tabela 3.1 e um campo *Severity* que está por enquanto vazio;

   (b) Após a inserção na base de dados, o item é igualmente colocado numa fila de trabalho criada para armazenar a informação de todos os *patches* recolhidos.

   Os itens da fila de trabalho ficam marcados como pendentes, ou seja, ainda serão alvo de processamento por parte de outros processos;

8. O *browser* é fechado, *CloseBrowser()*.

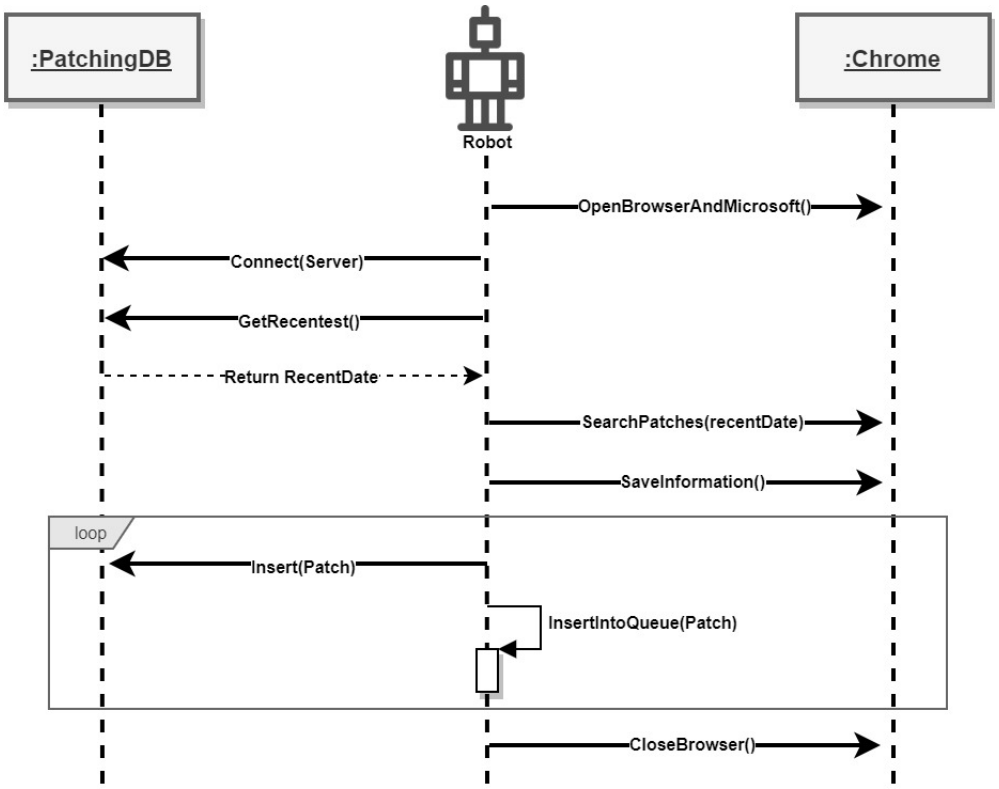



Figura 3.7: Representação do processo *Get Microsoft Updates List*.

| Campo | Descrição |
|---|---|
| *Date* | Data de lançamento do *patch* |
| *Article* | KB (identificador único do *patch* na *Knowledge Base*[5] da Microsoft) |
| *Type* | Tipo de *patch*[6] |
| *Applies To* | Versão do sistema operativo a que se aplica |

Tabela 3.1: Campos obtidos sobre cada *patch*.

[5]Uma *Knowledge Base* é um repositório centralizado onde informação é armazenada, organizada e partilhada.
[6]As atualizações podem ser *Monthly Rollup; Security Only; Security Update ou IE Cumulative*.

### 3.4.2 Obtenção das Vulnerabilidades

O *Common Vulnerabilities and Exposures* (CVE) é um dicionário que fornece definições de vulnerabilidades e exposições de cibersegurança divulgadas publicamente. O objectivo do CVE é facilitar a partilha de dados através de diferentes ferramentas, bases de dados e serviços. As entradas do CVE são constituídas por um número de identificação, uma descrição, e pelo menos uma referência pública [25].

As vulnerabilidades eram recebidas através de um *e-mail* mensal. Esse *e-mail* trata-se de uma *newsletter*, ilustrada na Figura 3.8 subscrita pelo CISO, onde o especialista de segurança em sistema operativo Windows, Randy Franklin Smith, resume a informação das vulnerabilidades de produtos Microsoft, divulgadas publicamente no mês corrente. O *e-mail* contém todos os detalhes para vulnerabilidades de severidade *Critical*, *Important* e *Moderate* presentes no *site* da Microsoft mas, de forma resumida e agrupada em forma de tabela. No processo *RoboCISO001 - 02 - Get CVE List From Outlook* a informação do *e-mail* é recolhida e inserida numa base de dados servindo depois para correlacionar cada vulnerabilidade com os *patches* que a mitigam.


| Technology | Products Affected | Severity | Reference | Workaround/ Exploited / Publicly Disclosed | Vulnerability Info |
|---|---|---|---|---|---|
| Windows | Windows 8.1, 8.1 RT, 10, Server 2012, 2016, 2019 | Critical | CVE-2020-0870<br>CVE-2020-1133<br>CVE-2020-1152<br>CVE-2020-1159<br>CVE-2020-1228<br>CVE-2020-1245<br>CVE-2020-1250<br>CVE-2020-1252<br>CVE-2020-1256<br>CVE-2020-1303<br>CVE-2020-1308<br>CVE-2020-1319<br>CVE-2020-16854<br>CVE-2020-16879 | Workaround: No<br>Exploited: No<br>Public: No | Security Feature Bypass<br>Elevation of Privilege<br>Remote Code Execution<br>Information Disclosure<br>Spoofing<br>Denial of Service |

Figura 3.8: Fragmento da tabela presente no *e-mail* da *Patch Tuesday*.

Este processo encontra-se representado na Figura 3.9 e é composto pelos seguintes passos:

1. O robô começa por aceder à sua caixa de correio e verifica se existem novos *e-mails* com o assunto *"Patch Tuesday"*, *GetCveEmail()*;

2. Se não existir nenhum *e-mail* a execução termina, caso contrário, o *e-mail* é guardado na máquina num ficheiro com extensão *.msg*, *SaveEmailAsFile(Email_id)*;

3. De seguida o *e-mail* é aberto através do terminal, de forma a ser possível ler o seu conteúdo, *OpenEmail(Email_id)*;

4. A tabela presente no corpo do *e-mail* é identificada e processada, colocando o seu conteúdo numa coleção, representada por *GetInfo()*;

5. O ficheiro e o terminal são fechados, *CloseEmailAndCmd()*;

6. O *e-mail* é marcado como lido na caixa de entrada e movido para a sub-pasta *processed*, onde são colocados todos os *e-mails* processados pelo RoboCISO, *MarkReadAndMove(Email_id)*;

7. Uma vez que já não é necessário, o ficheiro é apagado da máquina com a função *DeleteFile()*;

8. Tal como no processo anterior, é estabelecida uma ligação à base de dados, *Connect(Server)*;

9. A coleção resultante do processamento do *e-mail* é iterada:

   (a) Cada linha, ou seja, vulnerabilidade e os seus detalhes, é inserida na base de dados *PatchingDB*. A função representativa é *Insert(CVE)* e os campos inseridos estão descritos na Tabela 3.2;

   (b) Sempre que uma vulnerabilidade é inserida na base de dados, na tabela *CVE_List*, é igualmente inserida na fila de trabalho associada às vulnerabilidades, *InsertIntoCveQueue(Cve)*;

Após este passo, os itens da fila de trabalho estão marcados como pendentes e a execução termina.

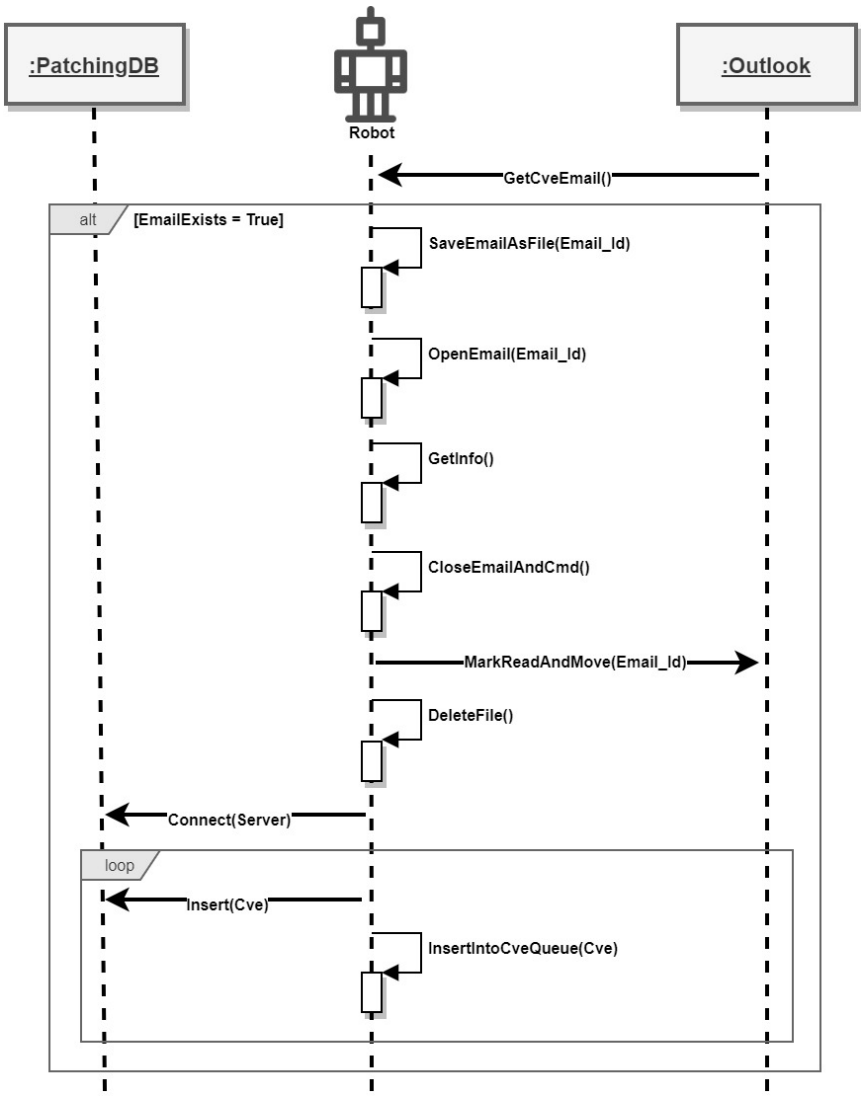



Figura 3.9: Representação do processo *Get CVE List from Outlook*.

| Campo | Descrição |
|---|---|
| *Products Affected* | Sistemas operativos que possuem esta vulnerabilidade |
| *Reference* | Identificador único da vulnerabilidade (*e.g.* CVE-2019-17613) |
| *Severity* | O nível de severidade da vulnerabilidade (*e.g. Critical*) |
| *Workaround* | Se existe alguma alternativa que pode ajudar a bloquear um ataque antes de fazer a atualização[8] |
| *Exploited* | Se a vulnerabilidade já foi explorada |

Tabela 3.2: Campos obtidos sobre a cada vulnerabilidade.

### 3.4.3 Obtenção dos patches para cada vulnerabilidade

Para cada vulnerabilidade, o *site* da Microsoft dispõe de uma página com os seus detalhes, nomeadamente com os *patches* necessários para a mitigar, organizados por versão de sistema operativo e contendo o nível de severidade atribuído pela própria Microsoft a cada um. Para cada vulnerabilidade presente na base de dados *PatchingDB* que apresente um nível de severidade *Critical*, será obtida toda essa informação. O funcionamento deste processo está ilustrado na Figura 3.10 sendo composto pelos seguintes passos:

1. O robô começa por obter os itens pendentes da fila de trabalho de vulnerabilidades, *GetPendingCvesQueue()*, guardando-os numa coleção;

2. É estabelecida a ligação à base de dados através da função *Connect(Server)*;

3. Iterando sobre a coleção de vulnerabilidades pendentes:

   (a) A primeira vulnerabilidade com uma severidade *Critical*, satisfazendo a condição ***CVE_Severity = Critical***, despoleta a execução do *launch* do Chrome, *OpenBrowser()* com o *url* dessa vulnerabilidade[9]. Este passo acontece apenas uma vez, evitando assim que se abra e feche uma janela do Chrome por item, o que iria aumentar bastante o tempo de execução. Para as restantes vulnerabilidades com severidade *Critical* o *url* é apenas atualizado com a vulnerabilidade corrente e a nova página é carregada, *GoToCVEPage(Cve)*;

   (b) Neste *site* existe uma tabela com a informação relativa a todos os *patches* que mitigam esta vulnerabilidade e que variam consoante a versão do sistema operativo que tratam. A informação dessa tabela é processada e inserida numa coleção, representada pela função *SaveInformation()*. Os campos recolhidos estão descritos na Tabela 3.3.

   (c) A coleção obtida é depois iterada:

      i. Cada linha da coleção contém um par (*Vulnerabilidade,Patch*) associado que é inserido na base de dados, na tabela *CVE_KB*, juntamente com a restante informação recolhida;

      ii. Cada linha é igualmente inserida na fila de trabalho de vulnerabilidades e *patches*, *InsertIntoCveKBQueue(Cve,Patch)*;

[8] https://docs.microsoft.com/en-us/security-updates/Glossary/glossary?redirectedfrom=MSDN#Workaround

[9] *https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/ + CVE*

(d) Se a vulnerabilidade possui uma severidade *Critical*, o *status* da sua entrada correspondente na fila de trabalho é atualizado e o item é marcado como *Completed*, *UpdateCVEStatus()*. Se a vulnerabilidade possui severidade *Important, Moderate* ou *Low*, o item é apenas marcado como *Completed*;

4. Após a iteração sobre todos os itens pendentes o Chrome é fechado, *CloseBrowser()*.

| Campo | Descrição |
|---|---|
| *Product* | Versão do sistema operativo |
| *Article* | Identificador único do *patch* |
| *Severity* | Severidade associada ao *patch* |
| *Supersedence* | *Patch* substituído |

Tabela 3.3: Campos obtidos sobre cada *patch*.

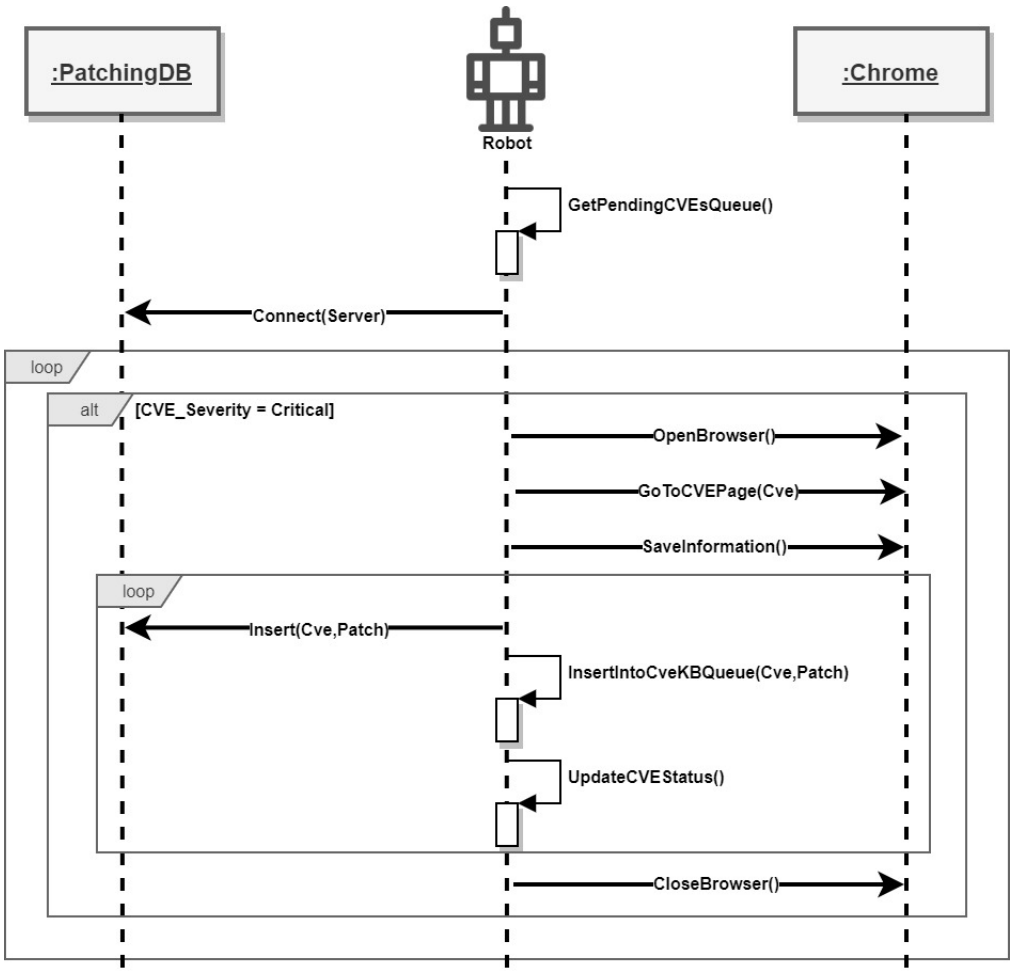



Figura 3.10: Representação do processo *Get KBs and Severity from each CVE*.

### 3.4.4 Classificação dos *patches*

Ficou definido que os *patches* serão classificados de acordo com o nível de severidade da vulnerabilidade a que estão associados. Após a obtenção dos *patches* que mitigam cada vulnerabilidade, é atribuído um

nível de severidade a cada *patch* obtido pelo primeiro processo executado, *RoboCISO001 - Get Microsoft Updates List*, que até aqui ainda não estava classificado. Comparando os *patches* provenientes desse sumário com os *patches* obtidos para cada vulnerabilidade, se existir uma correspondência, será possível atribuir um nível de severidade ao *patch* do sumário. Este processo está ilustrado na Figura 3.11 e é composto pelos seguintes passos:

1. É estabelecida a ligação à base de dados, *Connect(Server)*;
2. É executada uma *query* do tipo *UPDATE*, *ClassifyKBs()*, que irá verificar na base de dados a classificação associada ao *patch* na tabela *CVE_KB* e irá atribuí-la ao *patch* corresponde proveniente do sumário. A *query* utilizada nesta classificação está ilustrada na Secção A.1.
3. De seguida é feita uma *query* do tipo *SELECT* de forma a obter da tabela *KB_List* todos os *patches* inseridos nos últimos dois meses, que já possuem uma classificação no campo *Severity*. Este passo é representado pela função *GetKBList()* sendo o resultado uma coleção com o nome de *KB List*,
4. São obtidos os itens pendentes da fila de trabalho dos *patches*, *GetPendingKBs()*;
5. Iterando sobre a coleção *KB List*, proveniente da base de dados, atualiza-se a entrada correspondente da lista de itens pendentes, ou seja, que satisfazem a condição ***KB Classified = False***, não estando classificados, e portanto, o campo da severidade será atualizado e o item marcado como *Completed*.

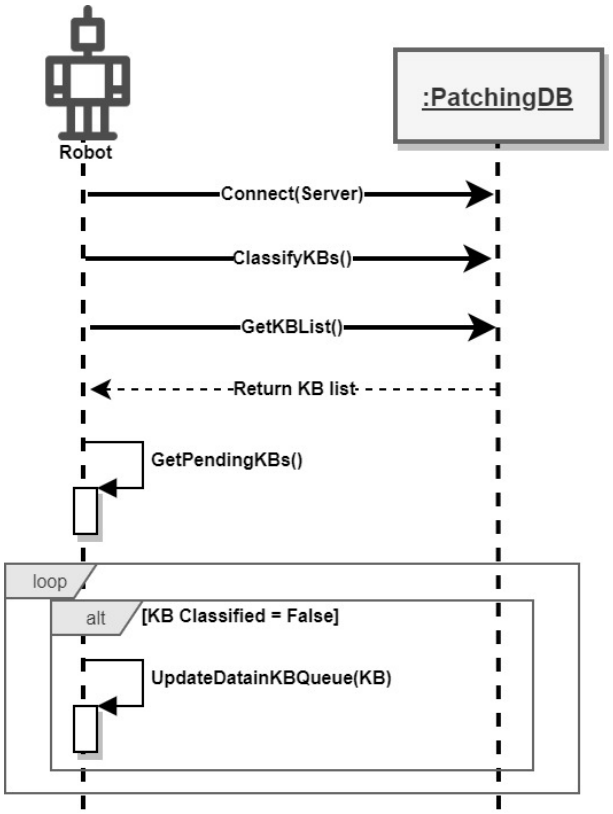



Figura 3.11: Representação do processo *Classify KBs*.

## 3.5 Arquitetura RoboCISO002 - Alerts

O *RoboCISO002 - Alerts* é o módulo responsável pela geração de todos os alertas e notificações gerados pelo RoboCISO, sendo composto por cinco processos. Todos os alertas/notificações são enviados via *e-mail* e mensagem WhatsApp. Cada alerta tem um *Id* único que é enviado como cabeçalho de cada mensagem e como assunto de cada *e-mail* permitindo facilmente associar cada *e-mail* à respetiva mensagem e vice-versa. Esse *Id* é composto pelo tipo do alerta, o código do alerta e o número do alerta. Na Tabela 3.4 está um exemplo de um possível *Id* de uma notificação de *SLA Exceeded*.

| | **Cabeçalho/Assunto** |
|---|---|
| WhatsApp | *SLA Exceeded #SLA000016* |
| E-mail | *RoboCISO - SLA Exceeded #SLA000016* |

Tabela 3.4: Exemplo de um *Id* de um *SLA Exceeded*.

O *e-mail*, para além de conter a informação enviada na mensagem, poderá em alguns casos conter informação complementar como acontece no *Patching Alert*. O utilizador primário do RoboCISO é o CISO, para quem todas as comunicações RoboCISO serão enviadas. Caso o CISO decida convidar outros elementos da sua estrutura governativa para receberem as comunicações do RoboCISO, estes são inseridos no grupo de WhatsApp e destinatários do *e-mail*.

### 3.5.1 RoboCISO002 - Patching Alert

O processo *RoboCISO002 - Patching Alert* é responsável por identificar instalações de *patches* em atraso e por gerar as respetivas notificações. Ficou definido que este processo é executado uma vez por dia às 9:10h da manhã. Este é o único processo que envia sempre uma notificação quer existam situações a reportar ou não, informando disso mesmo. Este é um vetor crítico pertencente à componente de **Atenção** e se for negligenciado pode resultar num novo ciberataque bem sucedido à organização.

A descrição está dividida em duas partes, a primeira focada na obtenção do conteúdo da notificação, ilustrada na Figura 3.12 e a segunda na geração e envio da mensagem e do *e-mail*, ilustrada na figura Figura 3.13. Se a instalação de um *patch* está em atraso para número total de máquinas superior a 100 e uma das seguintes condições for satisfeita, a notificação irá conter toda a informação relativa a esse *patch*:

- o *patch* foi lançado **há mais de 28 dias** e ainda é **requerido em mais de 10% das máquinas** que requerem esse *patch*, ou
- o *patch* foi lançado **há mais de 21 dias** e ainda é **requerido em mais de 25% das máquinas** que requerem esse *patch*, ou
- o *patch* foi lançado **há mais de 14 dias** e ainda é **requerido em mais de 50% das máquinas** que requerem esse *patch*.

Uma notificação pode conter informação relativa a vários *patches*. Caso não existam *patches* em atraso é gerada uma notificação que informa disso mesmo contendo a frase: "*For a total number of*

*assets >100, there are no required patches that reach the alert conditions in (1) or (2) or (3).*", seguida das condições de alerta enumeradas de (1) a (3).

Os passos da primeira etapa do processo são os seguintes:

1. O robô começa por obter o *Id* da nova notificação, *GetAlertID(PAlert)*.
2. Para aceder à base de dados do *Patching*, estabelece a ligação *Connect(Server)*;
3. É executada uma *query* do tipo *SELECT*, presente no Apêndice A, de forma a obter os *patches* que se encontram nas condições referidas anteriormente, *GetCriticalPatchList(Query)*;
4. É retornada a coleção de *patches*, *PAList*. Esta coleção contém para cada *patch* os campos especificados na Tabela 3.5;
5. Cada elemento da coleção é inserido na fila de trabalho de alertas.

| Campo | Descrição |
|---|---|
| *KB_Date* | Data em que o *patch* foi lançado |
| *Days Delay* | Dias que passaram desde que o *patch* foi lançado |
| *Patch* | Identificador único do *patch* |
| *Severity* | Severidade associada ao *patch* |
| *Patch Source* | Entidade que trata da gestão da instalação do *patch* na Altice |
| *Chassis Type* | Tipo da máquina se é servidor, *desktop* ou *laptop* |
| *# Installed* | Número de máquinas que têm o *patch* instalado |
| *# Required* | Número de máquinas que ainda requerem a instalação do *patch* |
| *# Total* | Número total de máquinas que requerem ou têm o *patch* instalado |
| *% Required* | Percentagem de máquinas que ainda requerem a instalação do *patch* |

Tabela 3.5: Campos da coleção *PAList*.

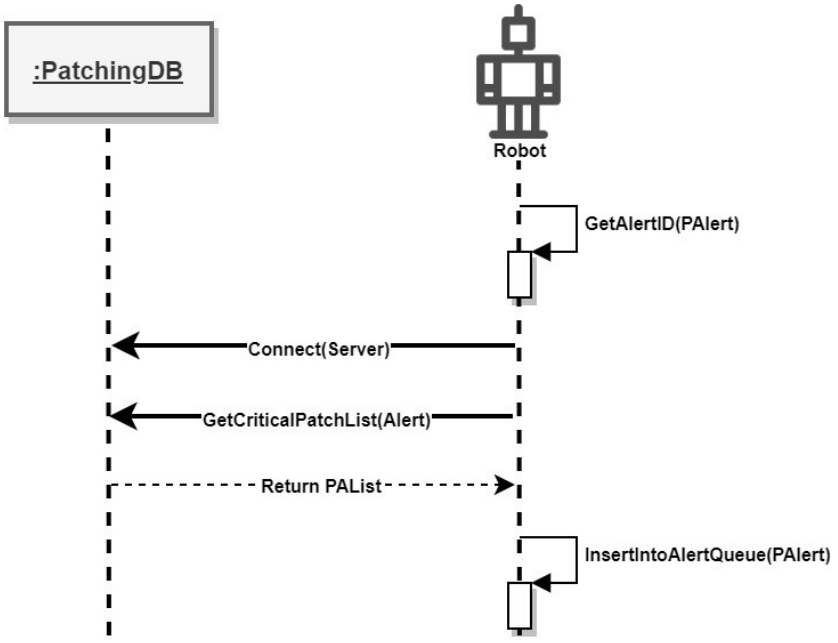



Figura 3.12: Representação da primeira parte do processo *Patching Alert*.

Na segunda fase, ilustrada na Figura 3.13, são executados os seguintes passos:

1. A coleção *PAList* é processada pela função *GenerateMailBody(PAAlertID,PAList)*, onde é gerado o corpo do *e-mail* da notificação a ser enviada. Este *e-mail*, caso existam *patches* em atraso, será composto por uma tabela com os campos anteriormente descritos, incluindo uma coluna que indica qual a condição alcançada para o *patch* estar a ser reportado, bem como uma coluna com o *link* para a página *web* do *patch*. Se não existirem atrasos a reportar o *e-mail* irá conter apenas um breve texto que informa que nenhuma das condições foi satisfeita.

2. A coleção é iterada e para cada *patch* a sua entrada na fila de trabalho é atualizada indicando que o *e-mail* foi enviado, *UpdateQueueStatus()*;

3. De seguida é aberto o Chrome na página do WhatsApp Web. Para a sessão estar iniciada na conta WhatsApp do RoboCISO foi necessário fazer manualmente o início da sessão uma vez. Este início de sessão fica válido para todos os processos e apenas será encerrada se feito o *logout* manual. A mensagem é enviada, *SendMessage(PAList, PAAlertID)*, para o grupo previamente criado;

4. Após o envio da mensagem o Chrome é fechado, *CloseWhatsappWeb()*;

5. Para cada elemento da coleção, o estado da respetiva entrada na fila de trabalho é atualizado indicando que a mensagem foi enviada, *UpdateQueueStatus()*;

6. Por fim, caso a coleção contenha elementos é inserida no HIDRA, *Insert(PAList)*.

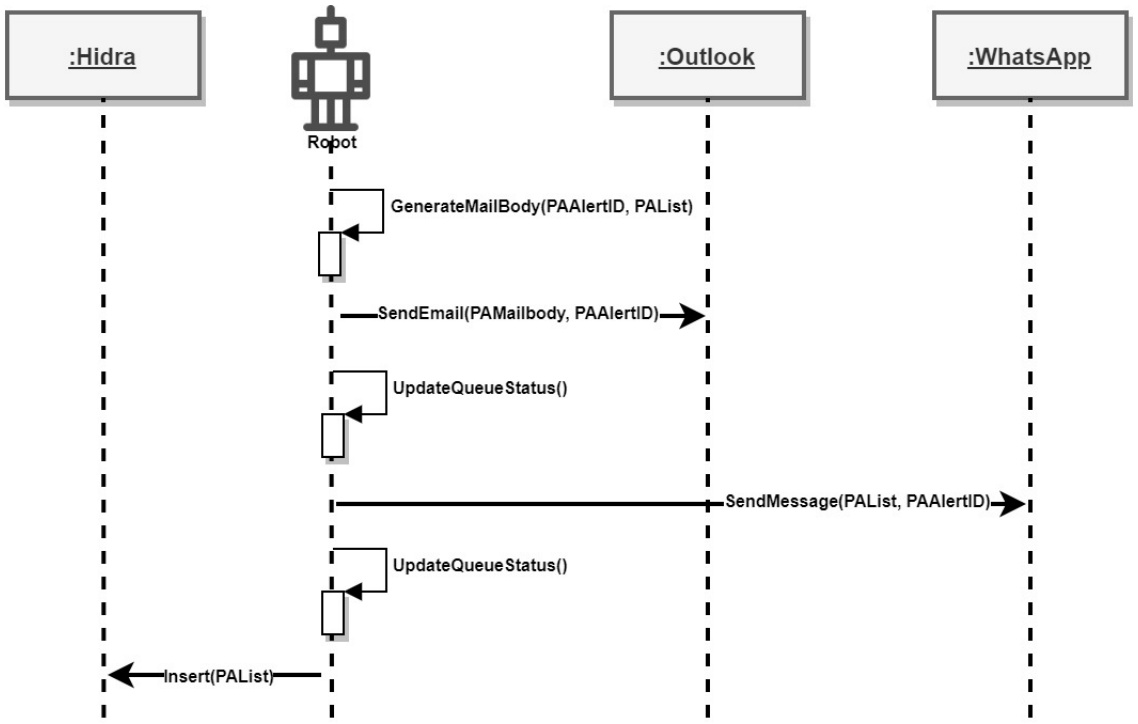



Figura 3.13: Representação da última parte do processo *Patching Alert*.

### 3.5.2 RoboCISO002 - DDoS Alert

O processo *RoboCISO002 - DDoS Alert* é referente aos alertas de ataques DoS/DDoS[9] que fazem parte da componente de **Urgência**. Uma das plataformas de deteção e mitigação de ataques de DoS/DDoS utilizada pela Altice Portugal é o Arbor Networks[10]. Quando um ataque é detetado por este sistema, é criado um *ticket* no RTIR, o sistema de *ticketing* utilizado pelo CSOC. Estes alertas são enviados sempre que existe um ataque que o Arbor considera relevante, ou seja, ultrapassa os limites definidos pela aplicação. Acontece que a grande maioria desses ataques não apresenta um volume nem uma duração suficientes que justifiquem a sua chegada ao CISO. Os parâmetros do Arbor são desta forma, demasiado baixos o que origina um excesso de informação, neste caso demasiados SMS.

A grande motivação para a criação do processo *RoboCISO002 - DDoS Alert* é diminuir o número de alertas que chegam ao CISO e que aqueles que chegam são realmente relevantes. Através da aplicação de valores limite que poderão ser afinados sempre que o CISO o entender, o excesso de informação que o CISO recebe diminui. Ficou estabelecido que este processo irá ser executado uma vez por hora, verificando a existência de novos *tickets*. A informação obtida através dos *tickets* é processada, sendo filtrada de acordo com os valores limite definidos:

- Se **a duração do ataque excede os 20 minutos e tem um volume máximo superior a 1 Gbps** (Gigabits por segundo) ou,
- Se **o volume máximo do ataque é superior a 5 Gbps**.

Podem resultar três alternativas desta execução:

- **O ataque iniciou e ainda não terminou**: *i.e.*, Neste caso será enviado um alerta via *e-mail* e uma mensagem via WhatsApp com a informação de que teve início um ataque de DoS/DDoS.
- **O ataque iniciou e terminou (ambos os passos detetados pelo robô)**: *i.e.*, o início do ataque foi reportado anteriormente pelo robô. Até o ataque ser dado como terminado será enviado um alerta com a informação de que este ainda está a decorrer. Quando o ataque é dado por terminado será enviado um alerta com essa informação.
- **O ataque iniciou e terminou (não detetado pelo robô)**: *i.e.*, o robô verifica que existiu um ataque que iniciou mas que terminou desde a sua última execução. Será enviado o alerta apenas com a informação de término do ataque uma vez que esta contém toda a informação necessária. Isto pode acontecer pois o robô ao ser executado uma vez por hora pode detetar ataques com duração superior a 20 minutos que ocorreram entre execuções e que entretanto já terminaram.

O caso de uso está ilustrado em duas partes, a primeira na Figura 3.14 e a segunda na Figura 3.15.

[9] Um ataque de DDoS acontece quando utilizadores legítimos não conseguem aceder a sistemas de informação, dispositivos ou outros recursos de rede devido a ações levadas a cabo por indivíduos com intenções maliciosas. Uma das formas mais comuns de realizadar um ataque de *dos* é aquela em que o atacante inunda a rede do servidor com pedidos ilegítimos que possuem endereços de retorno fabricados que levam o servidor a tentar responder [26].

[10] https://www.netscout.com/arbor-ddos

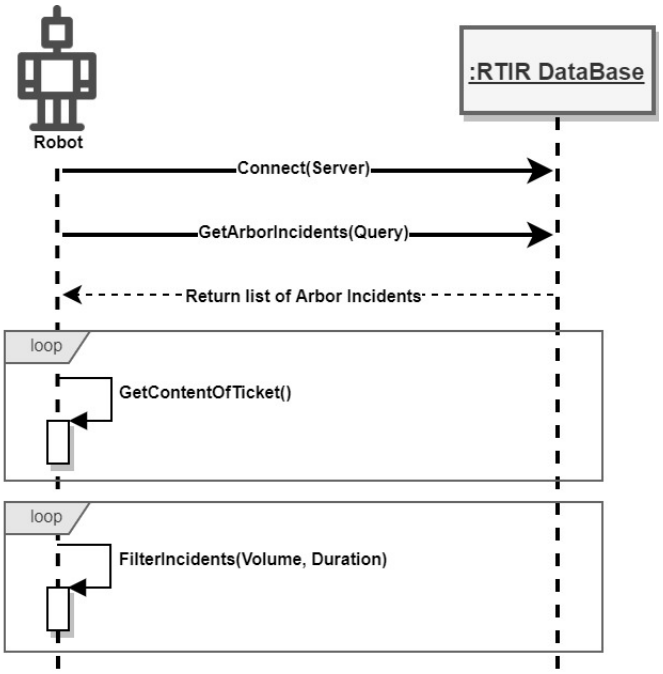



Figura 3.14: Representação da primeira parte do processo *DDoS Alert*.

A primeira parte do processo decorre da seguinte forma:

1. É estabelecida a ligação à base de dados do RTIR, *Connect(Server)*;

2. É executada uma *query* do tipo *SELECT* de forma a obter a informação dos novos *tickets* relativos a ataques DoS/DDoS registados, *GetArborIncidents(Query)*. O resultado vem sob a forma de uma coleção com os campos descritos na Tabela 3.6;

3. A coleção com essa informação é retornada, iterada e os seus campos são processados, *GetContentOfTicket()*. O volume do ataque é obtido através do corpo do *e-mail* do *ticket* e a duração do ataque é calculada:

   - Se o ataque ainda está a decorrer, o campo de data de resolução do *ticket* apresenta a data '1970-01-01 00:00:00', portanto o cálculo é realizado utilizando a data de criação do *ticket* e a hora local atual;
   - Se o ataque já terminou a duração é calculada com os valores presentes nos campos.

   Se a duração for inferior a 1h o resultado será apresentado em minutos, caso contrário em horas. Após o processamento do corpo do *e-mail* a coleção resultante será composta pelos campos apresentados na Tabela 3.7.

4. Após o processamento da informação, os ataques são filtrados de acordo com as condições de alerta especificadas anteriormente, *FilterIncident(Volume, Duration)*;

| Campo | Descrição |
|---|---|
| *Id* | Id do *ticket* |
| *Subject* | Assunto do *e-mail* |
| *Status* | Estado do ataque (*open, resolved*) |
| *IncidentClassification* | Classificação do incidente (DoS/DDoS) |
| *Resolution* | Se foi resolvido com sucesso |
| *Constituency* | Cliente afetado pelo ataque |
| *Created_Dt* | Data de criação do *ticket* |
| *Resolved_Dt* | Data de fecho do *ticket* |
| *MailBody* | O corpo do *e-mail* |

Tabela 3.6: Campos obtidos através do RTIR.

| Campo | Descrição |
|---|---|
| *RTIR Id* | Id do *ticket* no RTIR |
| *Subject* | Assunto do *e-mail* |
| *Max Volume* | Volume máximo do ataque (ex: 902.50 Mbps/87.59Kpps) |
| *Duration* | Duração do ataque em minutos |
| *Start Date* | Data de criação do *ticket* |
| *End Date* | Data de fecho do *ticket* |
| *Status* | Estado do ataque (*open, resolved*) |

Tabela 3.7: Campos finais para o *DDoS Alert*.

A segunda parte do processo foca-se na criação do alerta sendo composta pelos seguintes passos:

1. É obtido o *Id* do novo alerta, *GetAlertID()*;
2. Se a lista estiver vazia, o elemento inserido na fila de trabalho de Alertas indicará que não existem incidentes a reportar e a execução do processo termina, caso contrário, os incidentes existentes são inseridos, *InsertIntoQueue(Incidents, AlertID)*;
3. No caso de existirem incidentes, a coleção é processada pela função *GenerateMailBody(AlertID,-Incidents)* onde é gerado o corpo do *e-mail*, composto por uma tabela com os campos presentes na Tabela 3.7;
4. O *e-mail* é enviado através da função *SendEmail(Mailbody, AlertID)*.
5. O estado dos itens reportados é atualizado na fila de trabalho indicando que o *e-mail* foi enviado, *UpdateQueueStatus(Status)*;
6. Através do WhatsApp Web, o alerta é enviado para o grupo RoboCISO, *SendMessage(FilteredIncidents, AlertID)*;
7. O Chrome é fechado, *CloseWhatsappWeb()*;
8. O estado dos itens reportados é atualizado na fila de trabalho indicando que a mensagem foi enviada, *UpdateQueueStatus(Status)*;

9. Por fim o conteúdo do alerta enviado é inserido no HIDRA, *Insert(FilteredIncidents)*.

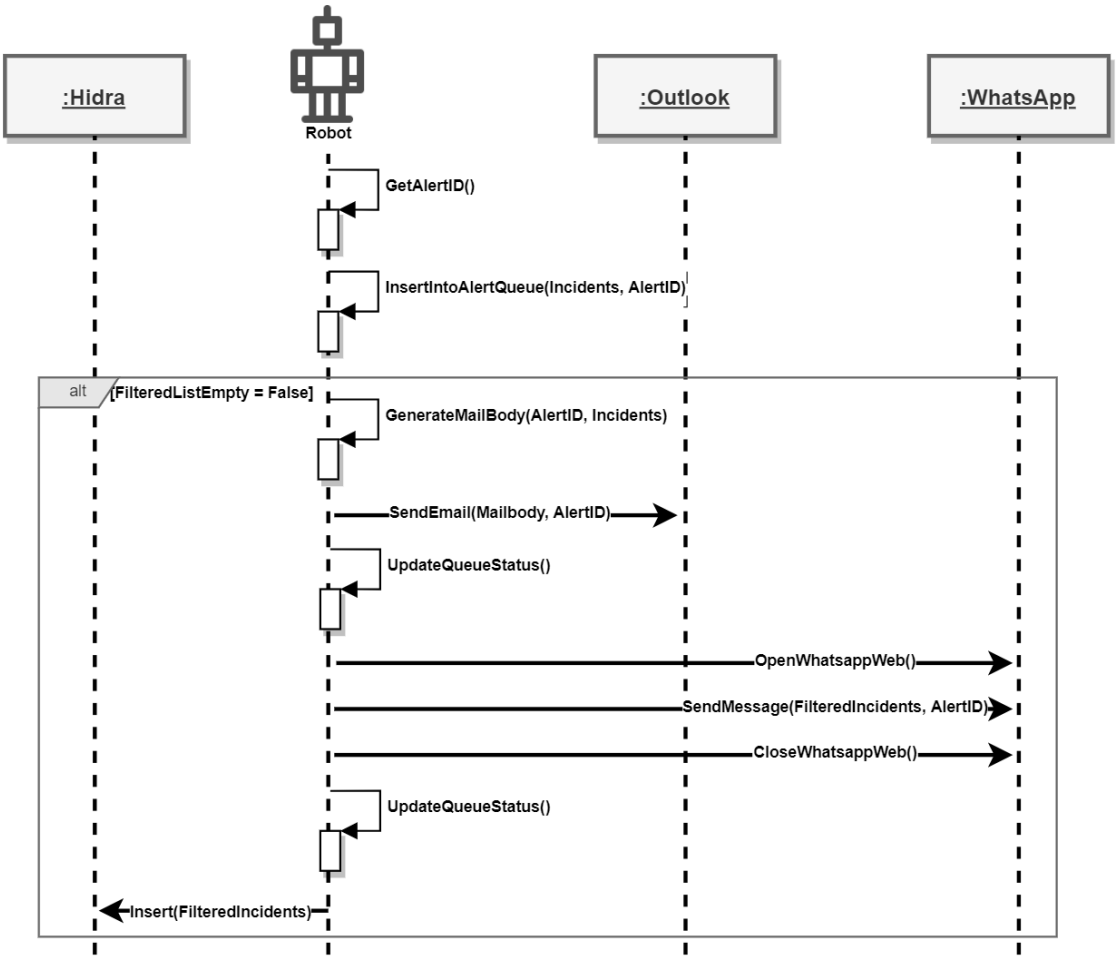



Figura 3.15: Representação da última parte do processo *DDoS Alert*.

### 3.5.3 RoboCISO002 - SLA Exceeded

No início deste projeto não existia um sistema que faça uma verificação do cumprimento ou incumprimento do SLA para tempo máximo de resolução dos *tickets* do RTIR. Para todos os eventos de segurança, que podem ser de diversos tipos como *phishing*, DoS/DDoS ou fraude, para os quais é aberto um *ticket* existe um SLA associado. A motivação deste caso de uso é informar o CISO em caso de incumprimento dos SLAs para incidentes internos assegurando assim que todos os *tickets* são efetivamente tratados.

Sendo este um processo da vertente de **Atenção** do CISO, foi estipulado que a sua execução diária é suficiente, sendo executado às 9:10h, depois do processo de *Patching Alert*. A informação obtida através da base de dados do RTIR será analisada e será calculado o tempo em que o SLA foi excedido e o tempo que demorou até à resolução do *ticket*. Aqueles que tenham **ultrapassado o SLA em mais de 2 horas** originarão uma notificação.

A primeira parte deste processo está ilustrada na Figura 3.16 e procede-se da seguinte forma:

1. É estabelecida a ligação à base de dados do sistema de *ticketing*, *Connect(Server)*;
2. É executada uma *query* do tipo *SELECT* obtendo a informação dos *tickets* registados na última semana, *GetRtirTickets(Query)*;

3. É devolvida a coleção de *tickets* que é iterada e os seus campos processados para calcular a duração na resolução do *ticket*, *CalcSLAExceeded(ticket)*:

   - Se o *ticket* ainda está aberto o campo de resolução do *ticket* apresenta a data '1970-01-01 00:00:00', portanto o cálculo da duração é feito utilizado a data de criação do *ticket* e a hora local atual;
   - Se o *ticket* já está fechado é calculada com os valores presentes nos campos.

   Nesta fase os *tickets* são também filtrados, se o SLA já tiver sido excedido em mais de duas horas o *ticket* passará à próxima fase do processo, caso contrário será descartado. A coleção que resulta deste processamento possui os campos descritos na Tabela 3.8.

| Campo | Descrição |
|---|---|
| *RTIR ID* | Id do *ticket* no RTIR |
| *Status* | Estado do *ticket* (*open*/*resolved*) |
| *Created Date* | Data de criação do *ticket* |
| *Subject* | Assunto do *ticket* |
| *Classification* | Tipo do incidente |
| *SLA* | Qual o SLA definido para o cliente |
| *Time To Solve* | Tempo que o *ticket* esteve aberto |
| *Diference* | Tempo em que o SLA foi excedido |

Tabela 3.8: Campos finais para a notificação *SLA Exceeded*.

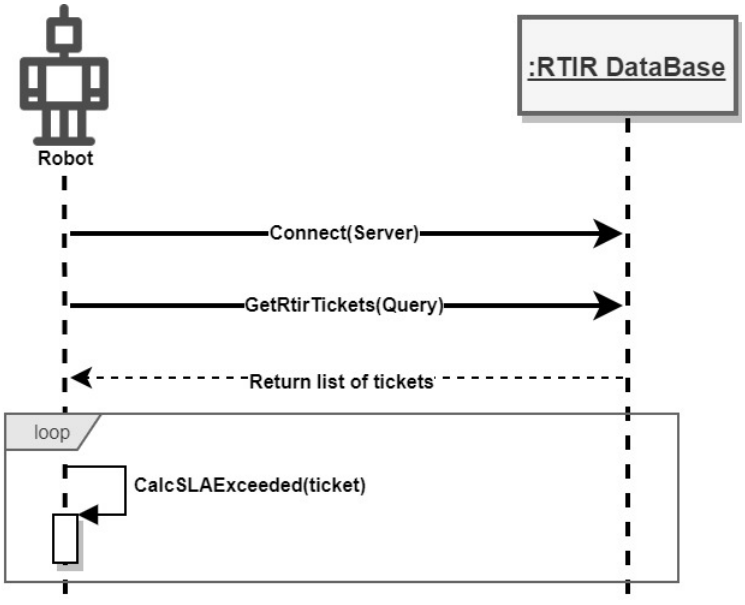



Figura 3.16: Representação da primeira parte do processo *SLA Exceeded*.

A segunda parte do processo focada no envio da notificação, está ilustrada na Figura 3.17 sendo composta pelos seguintes passos:

1. Após a filtragem, se a lista resultante estiver vazia a execução do processo termina, caso contrário, é obtido o *Id* da nova notificação, *GetAlertID()*;

2. A lista de *tickets* é processada pela função *GenerateMailBody(AlertID,ExceedTickets)*, onde é gerado o corpo do *e-mail*, composto por uma tabela com os campos descritos na Tabela 3.8;

3. O *e-mail* é enviado, *SendEmail(Mailbody, AlertID)*;

4. O estado dos itens reportados é atualizado na fila de trabalho indicando que o e-mail foi enviado, *UpdateQueueStatus(Status)*;

5. O robô acede ao WhatsApp Web, via Chrome, enviando a mensagem para o grupo RoboCISO, *SendMessage(ExceedTickets, AlertID)*;

6. O Chrome é fechado, *CloseWhatsappWeb()*;

7. O estado dos itens reportados é atualizado na fila de trabalho indicando que a mensagem foi enviada, *UpdateQueueStatus(Status)*;

8. A coleção é inserida no HIDRA, *Insert(ExceededTickets)*.

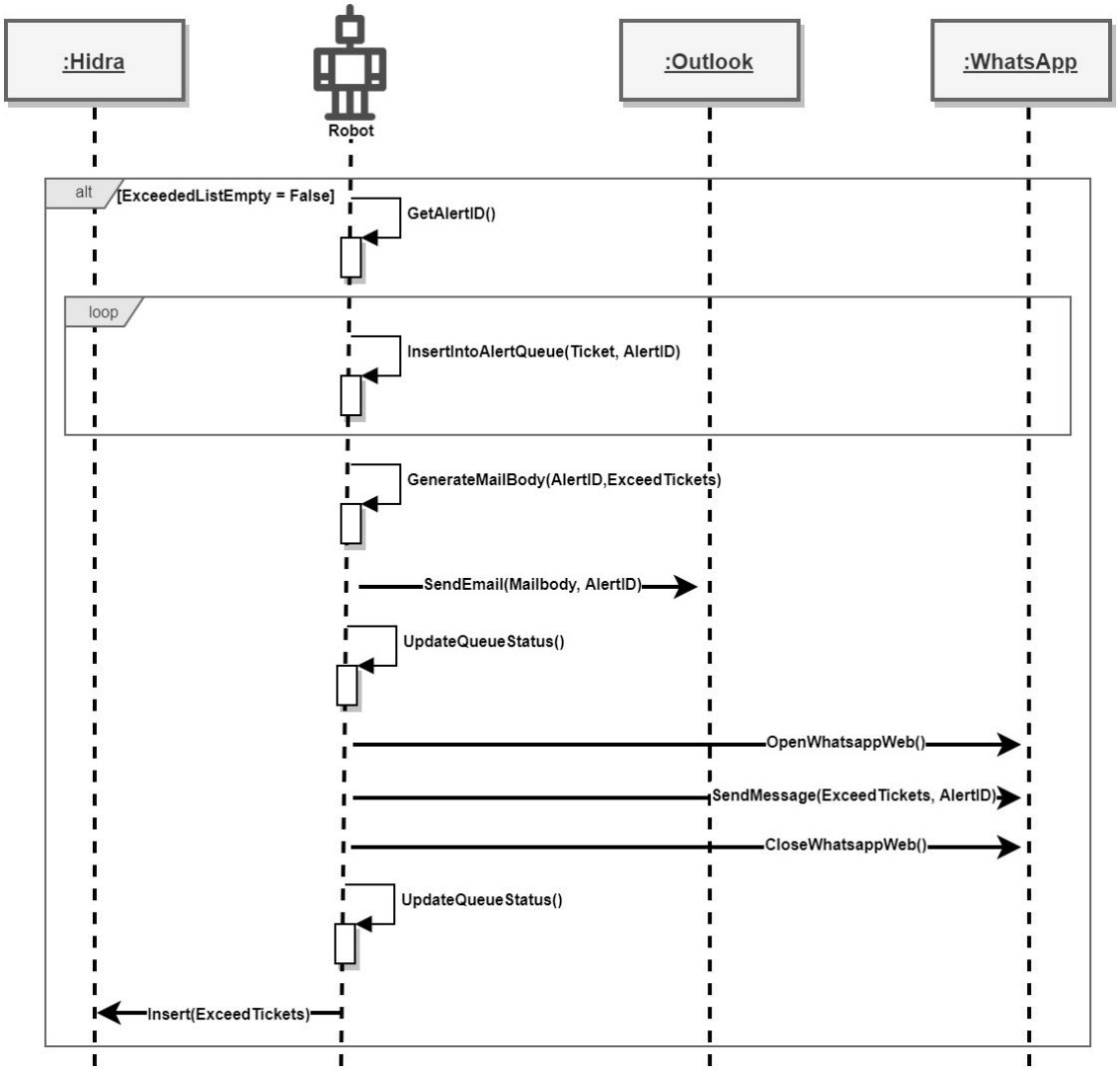



Figura 3.17: Representação da última parte do processo *SLA Exceeded*.

### 3.5.4 RoboCISO002 - BitSight Alert

A Bitsight[11] é uma solução de *security ratings*, que permite às organizações avaliarem o seu ciber risco de forma relativa à sua concorrência e aos seus *stakeholders*.

O valor dos *security ratings* varia entre 250 a 900, sendo que o mais elevado indica um melhor desempenho em termos de cibersegurança. Toda a informação recolhida pela Bitsight é acessível externamente, ou seja, não são feitos ataques maliciosos ou *pentest*[12] a nenhuma empresa para a obter essa informação. Os *security ratings* são calculados recolhendo e processando *terabytes* de informação de segurança de todo o mundo, que está distribuída em quatro categorias [27]:

- ***Compromised Systems***, referente aos dispositivos dentro da rede da organização que estão infetados com *malware*. Os sistemas comprometidos são identificados e classificados nos seguintes tipos de risco: *Botnet Infections*, *Spam Propagation*, *Malware Servers*, *Potentially Exploited* e *Unsolicited Communications*.

- ***Diligence***, referente aos registos que demonstram as medidas que a organização tomou para prevenir ataques. Os vetores de risco desta categoria são identificados e classificados tendo em conta alguns dos seguintes parâmetros: *Open Ports*, *TLS/SSL Certificates* e *Insecure Systems*.

- ***User Behavior***, que examina as atividades que podem introduzir *software* malicioso dentro da rede da organização, *i.e.* o *download* de um ficheiro comprometido.

- ***Public Disclosures***, que corresponde à recolha de informação acerca de quebras de segurança divulgadas publicamente e interrupções na continuidade de negócio, recorrendo para isso a uma grande variedade de fontes de notícias e serviços de agregação de quebras de segurança.

Os *ratings* são calculados de acordo com a seguinte fórmula:

$$CompromisedSystems \times 0.6 + Diligence \times 0.3 + UserBehavior \times 0.1$$

A verificação do estado dos *ratings* Bitsight está incluída na vertente de **Rotina** do CISO. Neste processo serão verificados os *ratings* de doze empresas do grupo Altice Portugal. Para facilitar esta atividade foi criado um processo que gerará alertas em caso de variações acentuadas num dado *rating*. Sempre que existem alterações nos *ratings* a Bitsight envia *e-mails* com essa informação. Na caixa de correio de um CISO estes podem facilmente passar despercebidos sendo necessária a análise dos *heatmaps* para perceber as variações ocorridas. Na Figura 3.18 está ilustrado um desses *e-mails*. Pequenas variações, de por exemplo 1%, não são relevantes o suficiente para gerarem um alerta. O limite que o CISO definiu como sendo de extrema importância ser avisado o quanto antes foi de variações acentuadas superiores a 5%, sejam estas positivas ou negativas. Para poder passar esta informação em tempo útil ao CISO ficou decidido que o processo será executado de forma horária, lendo a caixa de entrada do RoboCISO, para onde serão enviados os *e-mails* provenientes Bitsight com as alterações.

[11] https://info.bitsight.com/security-ratings-snapshot-request

[12] Ciberataque simulado autorizado a sistema informático, realizado para avaliar a segurança do sistema.

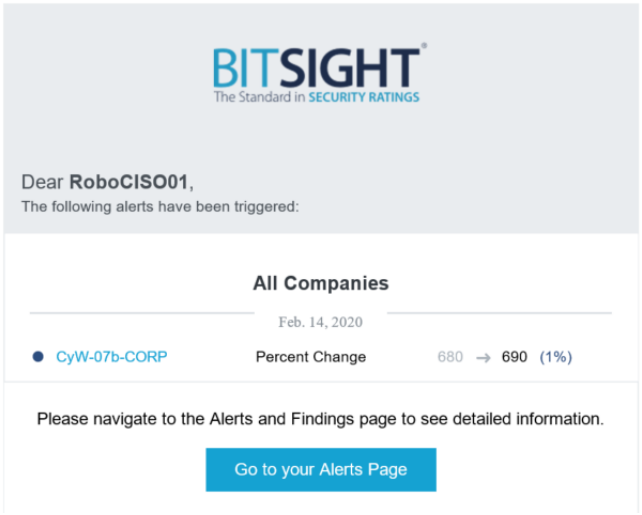



Figura 3.18: Exemplo de um *e-mail* de alerta enviado pela Bitsight.

A primeira parte do processo está ilustrado na Figura 3.19, sendo composta pelos seguintes passos:

1. O robô acede à sua caixa de correio e verifica se existem novos *e-mails* enviados pela Bitsight, *GetBitsightEmails()*;

2. Se não existirem novos *e-mails*, a execução vai para a função *InsertIntoAlertQueue(FinalColl, AlertID)* explicada no ponto 3 da segunda parte da execução. Neste caso ambos os parâmetros de entrada estarão vazios, dado que não existem novos *e-mails*. Existindo novos *e-mails* é criada uma coleção com os *Ids* dos *e-mails*. Esta coleção será iterada e cada *e-mail* será guardado na máquina num ficheiro com extensão *.msg*, *SaveEmailAsFile(Email_id)*;

3. Após todos os *e-mails* terem sido transferidos para a máquina são percorridos individualmente e são executados os seguintes passos:

   (a) O *e-mail* é aberto através do terminal, *OpenEmail(Email_id)*;

   (b) A informação do *e-mail* é obtida e processada, *GetInfo()*;

   (c) O *e-mail* é fechado, *CloseEmailAndCmd()*;

   (d) O *e-mail* é marcado como lido e movido para uma sub-pasta da caixa de entrada com o nome "*processed*", *MarkReadAndMove()*;

   (e) Após o processamento, o ficheiro é eliminado da máquina, *DeleteFile()*;

   (f) Cada *e-mail* pode conter mais do que um aviso de alteração de *rating*. Cada aviso é processado, transformado e inserido individualmente numa coleção temporária *Info*. Essa coleção temporária será inserida na coleção final *FinalColl* que irá conter todos os alertas recebidos, *AppentToCollection(Info)*;

   (g) O conteúdo da *FinalColl* é inserido na fila de trabalho que contém os avisos da Bitsight, *InsertIntoBitsightQueue(Collection)*.

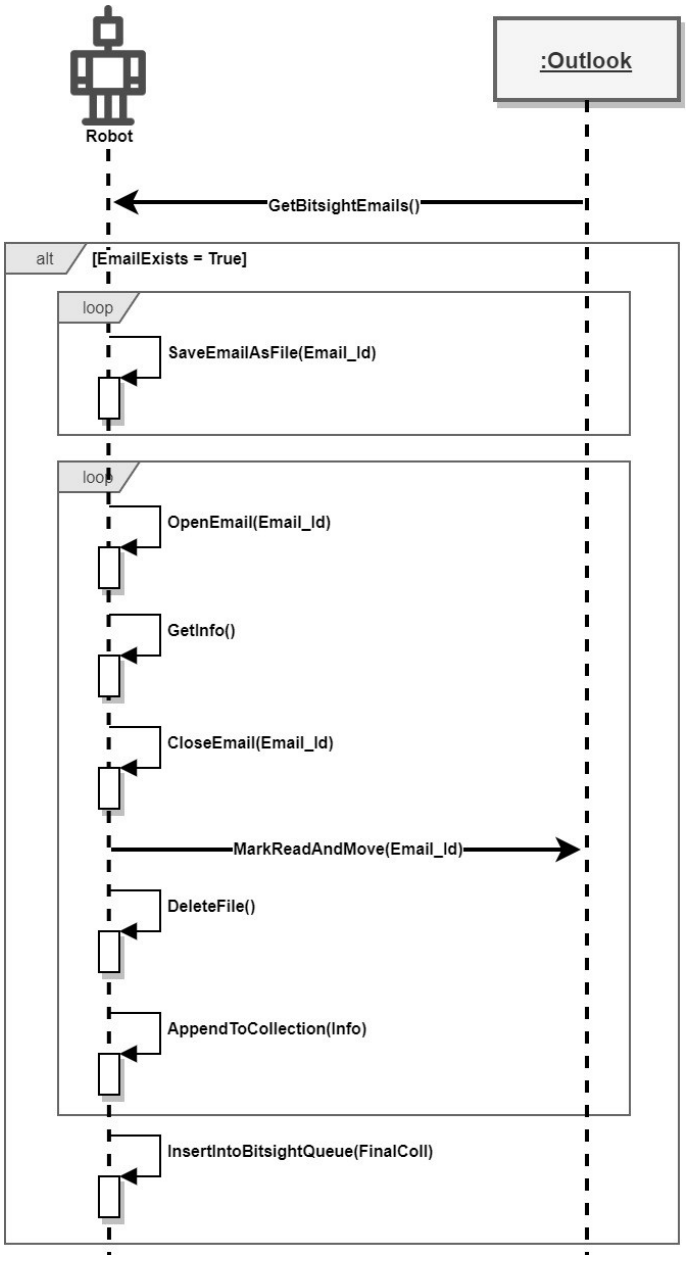



Figura 3.19: Representação da primeira parte do processo *Bitsight Alert*.

A segunda parte do processo, ilustrada na Figura 3.20, é composta pelos seguintes passos:

1. A coleção *FinalColl*, é iterada e é verificado se a alteração do *rating* em percentagem **é superior a 5% ou inferior a -5%**. Sempre que existir um item que satisfaça uma destas condições, será adicionado à coleção final, *AppendToFinalColl(Item)*, que irá conter os itens a serem reportados;

2. De seguida é verificado se a *FinalColl* possui elementos, se sim é gerado o *Id* do alerta através da função *GetAlertID()*. Se a *FinalColl* estiver vazia este passo não é executado;

3. Os elementos da *FinalColl* são inseridos individualmente na fila de trabalho de alertas, *InsertIntoAlertQueue(FinalColl, AlertID)*. Se a coleção estiver vazia, este é o último passo executado, sendo inserida uma entrada na fila de trabalho indicando que não existem alertas Bitsight;

4. O corpo do *e-mail* é gerado através da função *GenerateMailBody(AlertID,FinalColl)*;
5. O *e-mail* é enviado através da função *SendEmail(Mailbody, AlertID)*;
6. Os estados das respetivas entradas na fila de trabalho são atualizados, indicando que o *e-mail* foi enviado, *UpdateQueueStatus(Status)*;
7. Através do WhatsApp Web, via Chrome, a mensagem é enviada, *SendMessage(FinalColl, AlertID)*;
8. O Chrome é fechado, *CloseWhatsappWeb()*;
9. Os estados das respetivas entradas na fila de trabalho são atualizados, indicando que a mensagem foi enviada, *UpdateQueueStatus(Status)*;
10. O conteúdo do alerta enviado é inserido no HIDRA, *Insert(FinalColl)*.

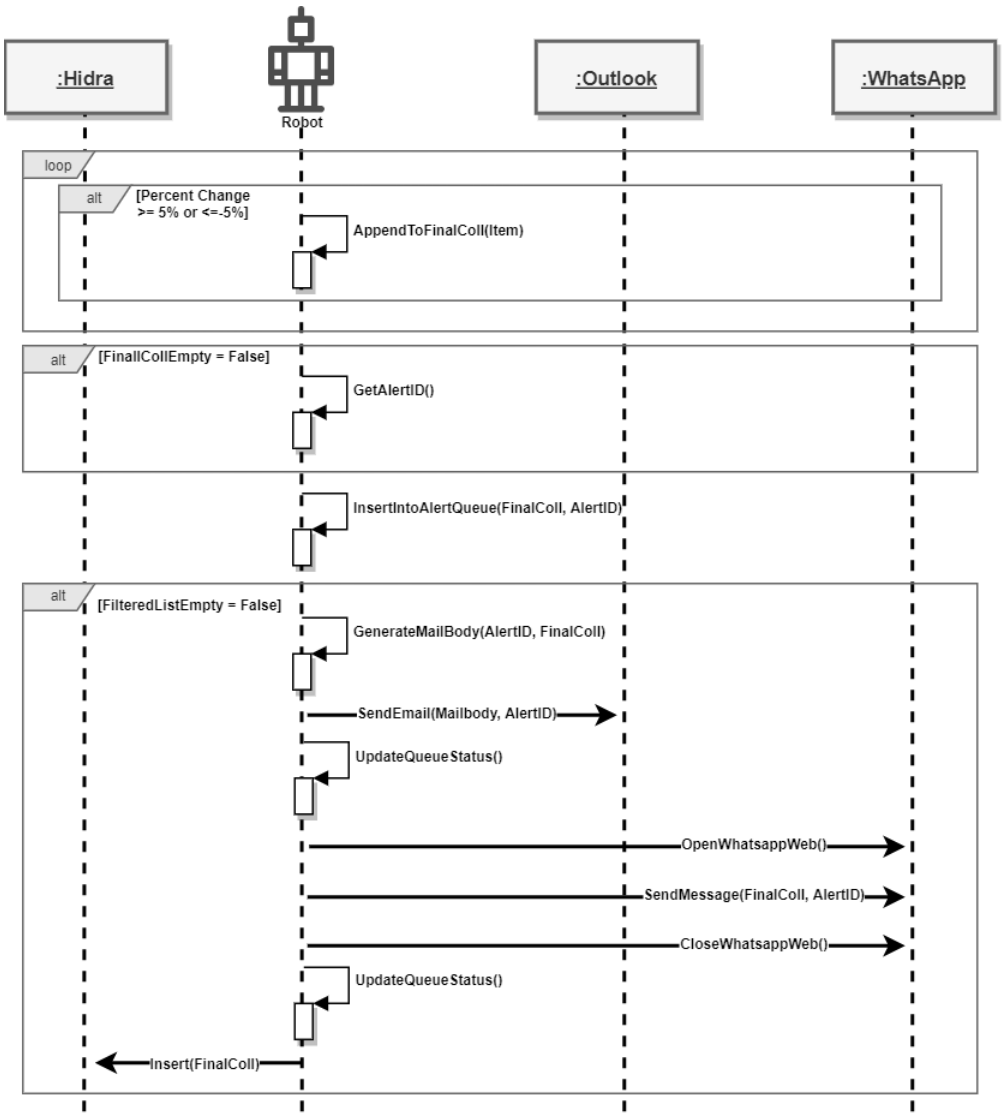



Figura 3.20: Representação da última parte do processo *Bitsight Alert*.

### 3.5.5 RoboCISO002 - Health Check

O processo *RoboCISO002 - Health Check* tem como objetivo a verificação do funcionamento de um conjunto de plataformas e componentes críticas, estando inserido na vertente de **Urgência**. Os eventos produzidos pelas diferentes fontes de informação utilizadas na organização, são armazenados por coletores e um dos destinos finais é o Alienvault, o SIEM. Um SIEM é a combinação das tecnologias de *Security Event Management* (SEM), utilizados para rever dados de *logs* e eventos provenientes das redes e sistemas da organização e, *Security Information Management* (SIM) utilizados para obter e reportar os dados dos *logs* [28]. A tecnologia SIEM fornece uma análise em tempo real de alertas de segurança sendo o seu principal objetivo auxiliar as organizações a responder a ataques de forma mais rápida [29]. Para cumprir este objetivo recolhe e agrega os dados dos *logs* das diferentes fontes permitindo a classificação e análise dos incidentes de segurança fornecendo relatórios e gerando alertas [30].

Uma lista de plataformas foi elaborada por um dos engenheiros de segurança da Altice Portugal, ordenada por prioridade. Foi realizado um levantamento do número de eventos registados por cada plataforma, ao longo de uma semana, de forma a perceber que plataformas possuem números suficientes para justificar a sua verificação. Foi possível concluir que diversas plataformas não possuem um número de eventos suficiente (*e.g.* 5 eventos por semana) que justifique o Health Check. Desta forma as plataformas/componentes verificados por este processo são as seguintes:

1. Componente de acesso remoto:
    - **VPN - Virtual Private Network**, alarga uma rede privada a uma rede pública e permite aos utilizadores enviar e receber dados através de redes partilhadas ou públicas como se os seus dispositivos informáticos estivessem diretamente ligados à rede privada da empresa, beneficiando das funcionalidades, segurança e gestão da rede privada.
2. Componente de proteção de autenticação dos postos de trabalho:
    - **DHCP - Dynamic Host Configuration Protocol**, um protocolo de gestão de rede utilizado em redes Internet Protocol (IP), através do qual um servidor DHCP atribui dinamicamente um endereço IP e outros parâmetros de configuração de rede a cada dispositivo para que estes possam comunicar com outras redes IP [31].
    - **NAC - Network Access Control**, visa controlar o acesso a uma rede com políticas e controlos sobre o local onde os utilizadores e os dispositivos podem aceder a uma rede e o que podem fazer [32].
    - **SuperCharger** - Plataforma de monitorização e gestão de autenticações em ambiente Windows [33].

Para além destas plataformas e componentes, que serão verificadas a partir do Alienvault, será verificado o estado do **HIDRA**, a base de dados não relacional onde são armazenados todo o tipo de eventos da organização. Esta plataforma é alimentada com processos ETL (extração, transformação e carregamento de dados) a partir de várias fontes relacionadas com a segurança, sendo o seu funcionamento baseado na integração de 3 tecnologias de base: *Elasticsearch*, *Kibana* [13], e *Logstash* [14] [6]. O *Kibana*

[13] https://www.elastic.co/products/kibana
[14] https://www.elastic.co/pt/logstash

é a interface de visualização dos dados e será a utilizada para verificar o seu funcionamento. A terceira plataforma cujo estado será verificado é o próprio **RoboCISO**, sendo o sistema que irá enviar todos os alertas e notificações, é de extrema importância que o seu funcionamento seja confirmado, pois a ausência de alertas pode não ser um sinal de que tudo está bem, mas sim, de que o sistema de alertas não está a funcionar. Ficou estipulado que este processo irá ser horário, tal como os processos *DDoS Alert* e *Bitsight Alert*. Devido ao número de eventos de cada plataforma/componente ficou definido que o Supercharger, NAC e VPN serão verificados de forma horária enquanto que o DHCP será verificado de forma diária numa das execuções do processo. O DHCP, para além de ser verificado numa base diária, não será verificado aos fins de semana uma vez que nesse período não são produzidos eventos. Este processo encontra-se descrito em três partes, sendo que a primeira está ilustrada na Figura 3.21, onde é feita a verificação dos sistemas que enviam eventos para o Alienvault, sendo composta pelos seguintes passos:

1. O robô começa por obter a hora local atual, definindo se se trata de uma verificação unicamente horária ou de uma verificação horária e diária. Se a hora atual se encontrar no período de verificação diária, definido como sendo entre as 10:15h e 10:45h trata-se de um processo horário e diário, satisfazendo a condição ***Hourly = False***, caso contrário é apenas horário ***Hourly = True***.

   Quando a execução do processo inicia existem duas coleções:

   - ***HourDatasource*** que contém as plataformas a verificar de forma horária.
   - ***DayDatasource*** contém as plataformas a verificar de forma diária.

2. Se a *flag Hourly = False* é verificado o valor de *Yesterday*: se for igual a 1 (domingo) ou 7 (sábado), o DCHP é removido das plataformas a verificar, uma vez que não existem eventos durante o período de fim de semana, caso contrário nada acontece;

3. É feito o *launch* do Chrome já na página de *login* do Alienvault, *OpenBrowserAndAlienvault()*;

4. O robô obtém as credenciais necessárias para fazer o *login*, sendo estas *user* e *pwd* e inicia a sessão na plataforma, *LoginAV(user,pwd)*;

5. De seguida vai até à página *Analysis* onde é possível procurar a informação sobre os eventos de cada componente/plataforma na última hora, *Last Hour* é a opção por omissão. Iterando sobre a coleção *HourDataSource* obtém para cada plataforma, os dados do último evento registado. Clicando no separador *Grouped*, obtém o número total de eventos na última hora. Aqui é criada uma nova coleção que contém todas as informações recolhidas. Nesta fase é também definido se a plataforma está ou não operacional verificado se o número de eventos da última hora é superior a zero. Se não for superior a zero, considera-se que a plataforma está em baixo e é atribuído *False* à *flag Working*, caso contrário atribui-se *True*;

6. Se a *flag* ***Hourly = False***, irá igualmente iterar sobre a coleção *DayDataSource*, definindo a procura por *Range*, em vez de *Last Hour*, e coloca a data do dia de ontem, *Yesterday*, tanto no campo *From* como no *To*. Após o carregamento estar concluído, recolhe a informação do último evento registado tal como no passo anterior. De seguida seleciona o separador *Grouped* onde recolhe o

número total de eventos registados pela plataforma nesse dia. Por fim atribui *True* se o número de eventos for superior a zero e *False* em caso contrário;

7. É feito o *logout* do AlienVault, *Logout()* e o Chrome é fechado, CloseBrowser().

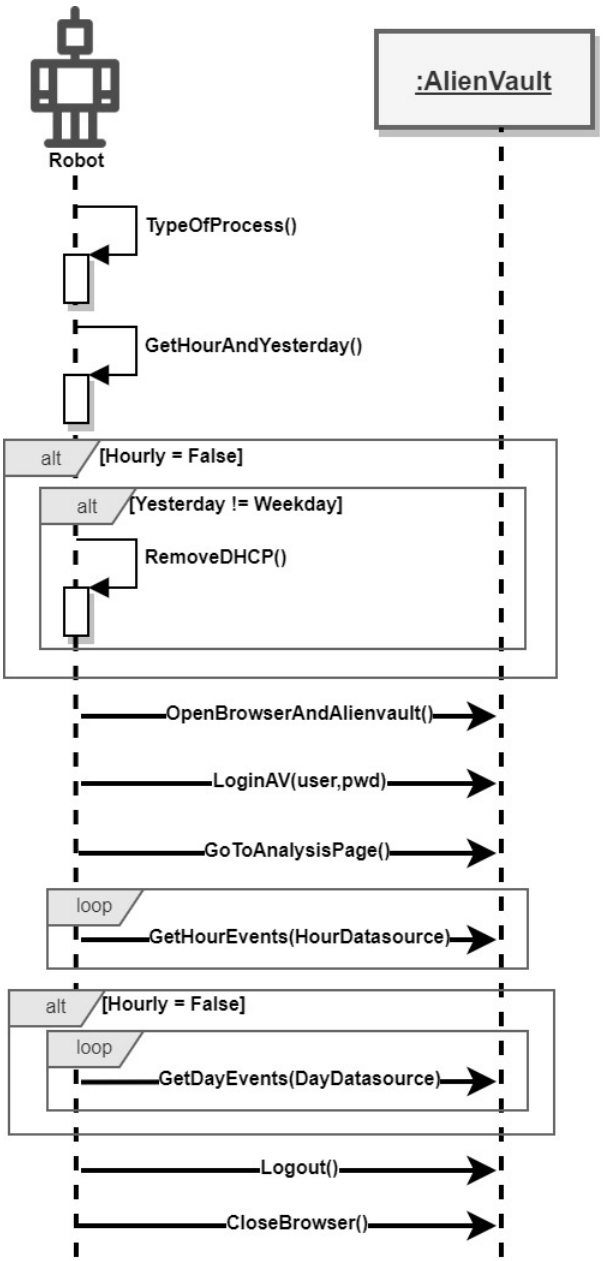



Figura 3.21: Representação da primeira parte do processo *Health Check*.

Após a interação com o Alienvault estar concluída o próximo passo é a verificação do estado do HIDRA e do RoboCISO, ilustrada na Figura 3.22, sendo efetuados os seguintes passos:

1. É feito o *launch* do Chrome na página inicial do Kibana , onde se faz a visualização dos dados do HIDRA, *OpenBrowserAndKibana()*;

2. De seguida seleciona a página *Discovery*, representada pela função *GoToDiscoverPage()*, que

contém os últimos eventos do tipo *IAM*, *Identity Account Management*, que correspondem a eventos de *login/logout* nas contas, VPN e aplicações;

3. A data do último evento e o número total de eventos registados nos últimos 15 minutos são guardados numa coleção. Se o **número de eventos for zero** considera-se que o HIDRA não está a funcionar e a *flag Working* é marcada a *False*. Este passo é representado no diagrama pela função *ReadLastEntryAndHist()*;

4. Após a recolha o *browser* é fechado, *CloseBrowser()*;

5. O processo de verificação do RoboCISO inicia-se obtendo todos os itens pendentes e todos itens completados na última hora da fila de trabalho de alertas. Os itens pendentes correspondem às entradas inseridas pelos processos *Bitsight Alert* e *DDoS Alert* sempre que não reportam alertas.

6. A coleção de itens completados é iterada e a data em que o item foi inserido na fila de trabalho é guardada numa variável designada de *Recent* e a informação sobre esse evento é guardada numa coleção designada de *Recent Collection*. O conteúdo da variável e da coleção será atualizado sempre que um item possuir uma data mais recente;

7. A coleção de itens pendentes é iterada e:

   (a) Cada item pendente tem o seu estado atualizado, indicando que foi verificado pelo *Health Check* e é marcado como *Completed*;

   (b) Se a data em que foi inserido na fila de trabalho é mais recente do que a guardada na variável *Recent*, esta é atualizada bem como o conteúdo da coleção *Recent Collection*.

8. Se a data do último evento colocado na fila de trabalho tem menos de 1 hora, considera-se que o RoboCISO está a funcionar como esperado e a *flag Working* é colocada a *True*, caso contrário, é colocada a *False*;

9. Após todas as plataformas estarem verificadas inicia-se a fase de geração de alertas onde se obtém o *Id* do novo alerta *Health Check*, *GetAlertID()*;

10. Toda a informação recolhida sobre as diferentes plataformas e componentes é colocada na fila de trabalho dedicada ao *Health Check*. Este passo é representado pela função *InsertIntoHcQueue(HIDRA, Rc, AVDaily, AVHour, AlertID)*, cujos parâmetros correspondem às coleções das plataformas: HIDRA, RoboCISO, Eventos diários do Alienvault, Eventos horários do Alienvault e o *AlertID*;

11. O último passo desta fase do processo consiste em iterar sobre os itens colocados na fila de trabalho e executar uma das seguintes ações:

    - Se a plataforma/componente está marcada como estando a funcionar (*Working = True*) é marcado como *Completed*.
    - Se a plataforma/componente está marcado como não estando a funcionar (*Working = False*) é marcado como *Completed* e adicionado à fila de trabalho de alertas;

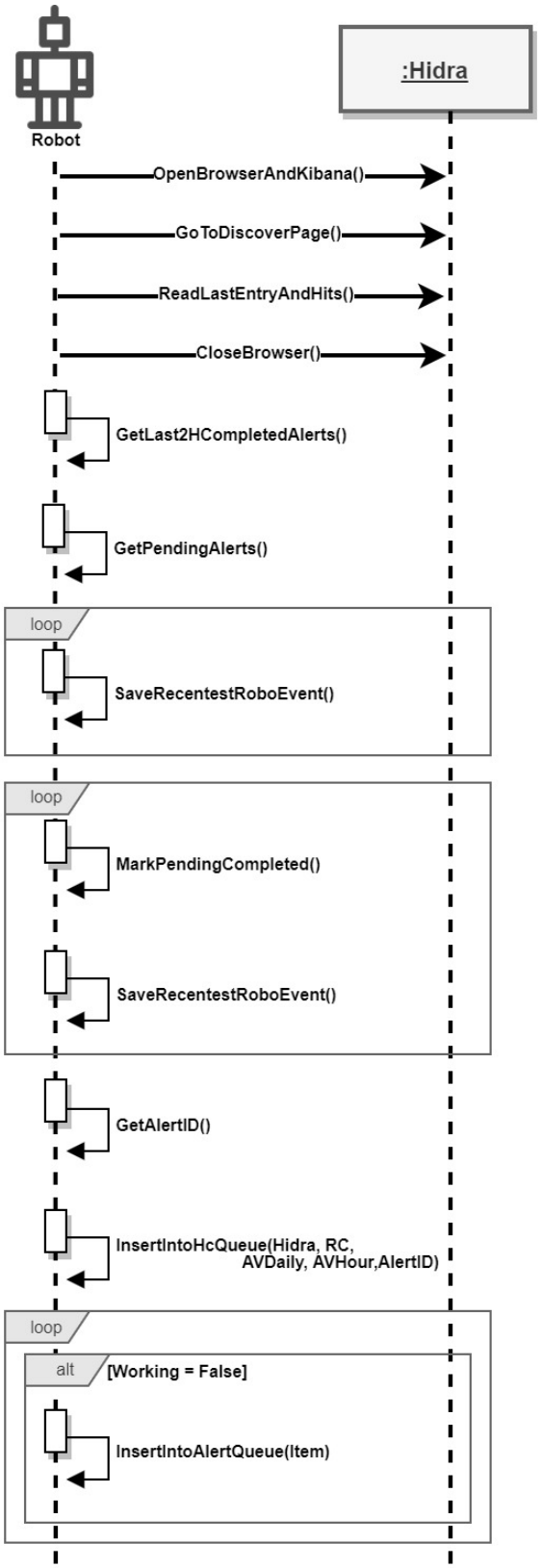



Figura 3.22: Representação da segunda parte do processo *Health Check*.

A última parte deste processo, ilustrada na Figura 3.23, é semelhante à dos restantes processos na medida em que os alertas são obtidos da fila de trabalho e enviados por *e-mail* e Whatsapp. Esta parte do processo é composta pelos seguintes passos:

1. A partir da fila de trabalho de alertas são obtidos os itens referentes ao processo *Health Check*. Nesta fase as coleções *AVDaily* e *AVHour* são fundidas numa só designada de *AVPlatforms*;
2. Se todas as coleções obtidas no passo anterior estiverem vazias, significa que não existem plataformas/componentes com problemas e a execução do processo termina. Se pelo menos uma das coleções apresentar elementos a execução continua e o corpo do *e-mail* é gerado através da função *GenerateMailBody(HIDRA, RC, AVPlatforms, AlertID)*;
3. O *e-mail* é enviado através da função *SendEmail(Mailbody, AlertID)*;
4. O estado dos itens na fila de trabalho é atualizado indicando que o *e-mail* foi enviado, *UpdateQueueStatus(Status)*;
5. Através do WhatsApp Web o alerta é enviado para o grupo RoboCISO, *SendMessage(HIDRA, RC, AVPlatforms, AlertID)*;
6. Após o envio da mensagem o Chrome é fechado, *CloseWhatsappWeb()*;
7. O estado dos itens na fila de trabalho é atualizado indicando que a mensagem foi enviada, *UpdateQueueStatus()*;
8. Os itens são inseridos no HIDRA, *Insert(HIDRA, RC, AVPlatforms)*.

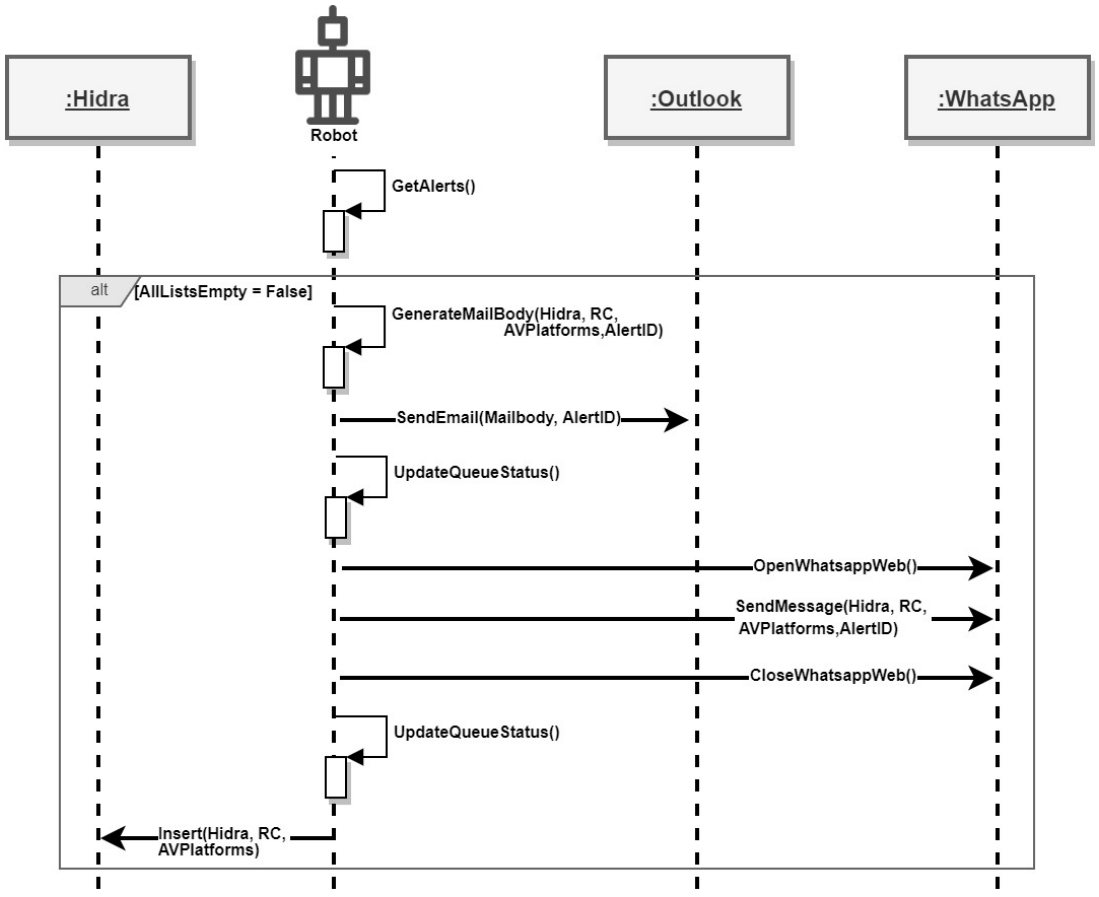



Figura 3.23: Representação da última parte do processo *Health Check*.

# Capítulo 4

# Implementação RoboCISO

Neste capítulo é descrita, em detalhe, a implementação dos processos que compõem os módulos *RoboCISO001 - Critical Patching* e *RoboCISO002 - Alerts*. Estes módulos foram implementados recorrendo ao Blue Prism, sendo esse o seu *core*, o Chrome foi o *browser* escolhido para a interação com os diferentes *sites* e aplicações.

## 4.1 Objetos Auxiliares

Para o funcionamento dos processos foi necessária a criação de objetos (VBOs[1]) para interagir com as aplicações necessárias. Os três VBOs desenvolvidos são utilizados em ambos os módulos.

- **RoboCISO001 - Chrome Interaction**, responsável por toda a interação com o *browser* Chrome, utilizado no módulo *RoboCISO001 - Critical Patching* na obtenção dos *patches* e vulnerabilidades e no processo *RoboCISO002 - Health Check* para aceder ao Kibana. Este objeto contém as seguintes ações[2] implementadas:
  - **Launch** - Faz o *launch* do Chrome.
  - **Attach** - Garante a conexão do Chrome ao BP para que este consiga identificar os elementos necessários para a execução do processo.
  - **Terminate** - Fecha o Chrome deixando o ambiente no seu estado inicial.
  - **Get List Of Updates** - No processo *RoboCISO001 - 01 - Get Microsoft Updates List* obtém a lista de atualizações presentes no site da Microsoft.
  - **Get KB from CVE** - No processo *RoboCISO001 - 03 - Get KBs and Severity from each CVE* obtém a lista de *patches* presentes no site da Microsoft para cada CVE.
  - **Go To Kibana** No processo *RoboCISO002 - Health Check* é utilizado para aceder ao *site* do Kibana e recolher a informação necessária.
- **RoboCISO001 - Outlook Interaction**, responsável pela interação com a aplicação do Outlook. Este objeto contém as seguintes ações implementadas:

[1] *Visual Business Object* (VBO) corresponde aos objetos BP, que têm como função implementar um conjunto de operações numa determinada interface de utilizador

[2] Uma ação no BP corresponde a uma das páginas do VBO. Cada página desempenha uma dada operação na aplicação alvo

  - **Configure Outlook** - Faz o *launch* do Outlook.
  - **Attach** - Garante a conexão do Outlook com o BP para que este consiga identificar os elementos necessários para a execução do processo.
  - **Get Table** - No processo *RoboCISO001 - 02 - Get CVE_list from Outlook* obtém a tabela de vulnerabilidades presentes num dado *e-mail*.
  - **Get Bitsight Info** - No processo *RoboCISO002 - BitSight Alert* obtém a informação presente num dado *e-mail*.
  - **Close Email** - Nos processos *RoboCISO001 - 02 - Get CVE_list from Outlook* e *RoboCISO002 - BitSight Alert* é utilizado para fechar os *e-mails* que têm de ser abertos individualmente para serem processados.

- **RoboCISO001 - WhatsApp Interaction** Apesar da interação com o WhatsApp ser efetuada via Chrome, foi criado um novo objeto dedicado a esse fim. Este objeto contém as seguintes ações implementadas:
  - **Launch**, **Attach** e **Terminate** - Explicado no objeto *RoboCISO001 - Chrome Interaction*;
  - **Open RoboCISO Conversation** - Abre a conversa do grupo "RoboCISO"no WhatsApp Web para onde são enviados os alertas.
  - **Send Message [SLA]** - Envia a mensagem do processo *RoboCISO002 - SLA Exceeded*.
  - **Send Message [DDoS]** - Envia a mensagem do processo *RoboCISO002 - DDoS Alert*.
  - **Send Message [Patching]** - Envia as mensagens do processo *RoboCISO002 - Patching Alert*.
  - **Send Message [Bitsight]** - Envia a mensagem do processo *RoboCISO002 - BitSight Alert*.
  - **Send Message [HCheck]** - Envia a mensagem do processo *RoboCISO002 - Health Check*.

## 4.2 RoboCISO001 - Critical Patching

O módulo *RoboCISO001 - Critical Patching* é composto por quatro processos cuja implementação é descrita nas próximas secções.

### 4.2.1 RoboCISO001 - 01 - Get Microsoft Updates List

O processo *RoboCISO001 - 01 - Get Microsoft Updates List* tem como objetivo a recolha dos *patches* lançados pela Microsoft no último mês. A aplicação utilizada por este processo é o Chrome. A sua implementação, ilustrada na Figura 4.1 sob a linguagem própria do Blue Prism, está dividida em três partes: ***Launch*** (1), ***Get Updates List*** (2) e ***Terminate*** (3).

Para cada passo na execução existe também uma componente de *Recover* associada, que permite que o processo lance uma exceção caso falha, o que facilita na deteção e na resolução do problema. O ***Launch*** do Chrome (1) corresponde a um dos objetos auxiliares criados, tendo como finalidade abrir o *browser* no *site* do *Microsoft Security Update*[3], permitindo a pesquisa dos *patches* que está implementada na página ***Get Updates List*** (2), cujo conteúdo está detalhado na Figura 4.2.

[3] `https://portal.msrc.microsoft.com/en-us/security-guidance/summary`

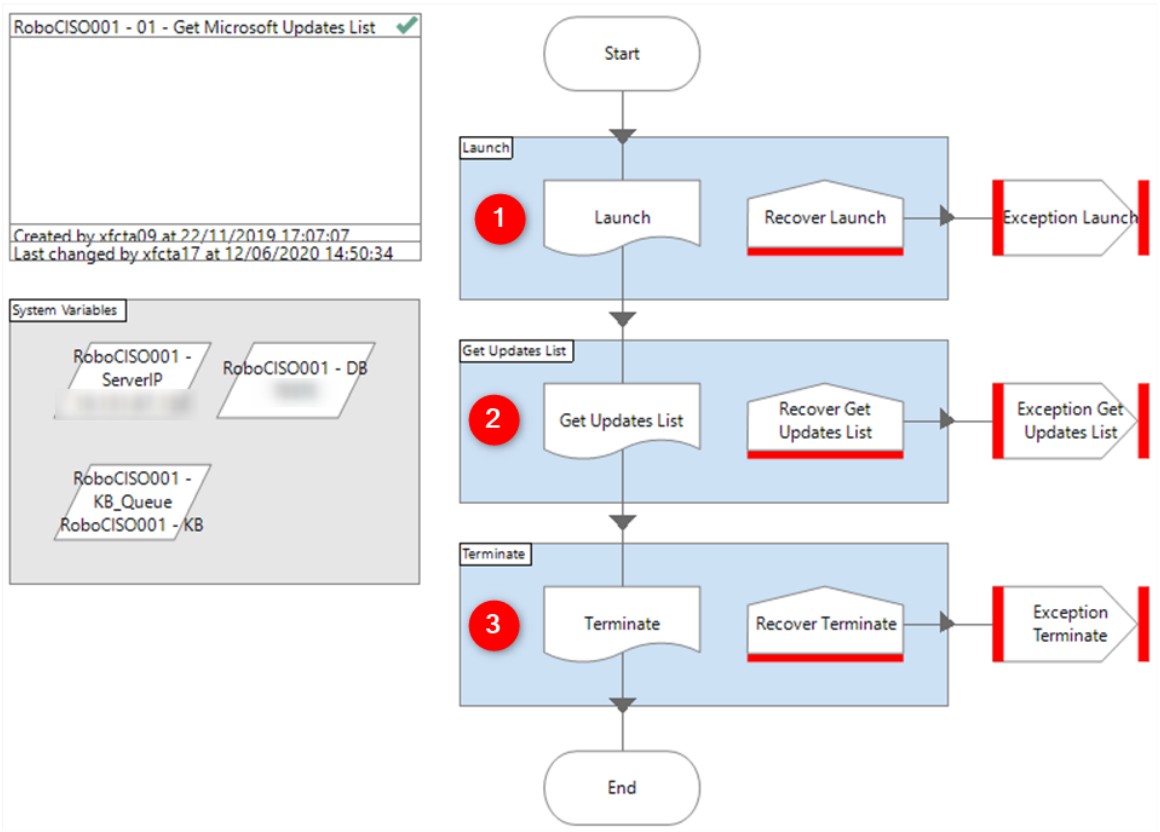



Figura 4.1: *Main Page* do processo *RoboCISO001 - 01 - Get Microsoft Updates List*.

A primeira ação da página ***Get Updates List*** (2) é a ligação à base de dados e obtenção da data mais recente aí presente, ***Get Most Recent Date*** (4), na Figura 4.2, guardando-a na variável ***Recent Date***. De seguida começa a interação com o Chrome, utilizando um dos objetos auxiliares criados ***Get List Of Updates*** (5). De seguida, a ***Recent Date*** é colocada no campo *From*, presente na página e que define a data a partir da qual serão pesquisados *patches*. O campo *To* mantém-se inalterado, uma vez que contém a data atual. Após a atualização da página, serão obtidos os novos *patches* que estão organizados sob forma de tabela. Cada linha da tabela é processada individualmente e inserida numa coleção, *Final Collection*. Essa coleção é iterada sendo efetuadas duas fases de validação aos seus itens antes da sua inserção na base de dados:

- ***!= KB?*** (6), verifica se o item corresponde realmente a um *patch*. Esta verificação é necessária pois muitas vezes o campo *Article*, que corresponde ao *Id* do *patch* pode estar vazio. Isto acontece pois no sumário das atualizações podem estar incluídas algumas notas e que nesse caso não possuem nenhum *patch* associado. Desta forma, se for um *patch*, a *query* para inserção na base de dados é atualizada com os dados do elemento iterado e, uma coleção temporária designada *Temp*, é preenchida com as informações do *patch*, *Update Query and Temp* (7). Se não for um *patch* passará ao próximo elemento;

- ***Item in Queue?*** (8), verifica se o item já se encontra na fila de trabalho *RoboCISO001 - KB*, a fila que contém a informação dos *patches*. Se já estiver, significa que também estará na base de dados e portanto não será inserido de novo, caso contrário, será adicionado a ambas. Para ser possível

ver de imediato alguma informação sobre o item, o BP permite a definição de alguns parâmetros, que são visíveis ao aceder à interface da fila de trabalho. Esses parâmetros são *Item Key, Status* e *Tag*, estando neste caso com os seguintes valores atribuídos:

- *Item Key*, corresponde ao identificador único do *patch*, o seu KB, por exemplo: 4556799;
- *Status*, após ser inserido na base de dados está definido como ***'01 - Inserted into DB'***;
- *Tag*, que neste processo está vazia.

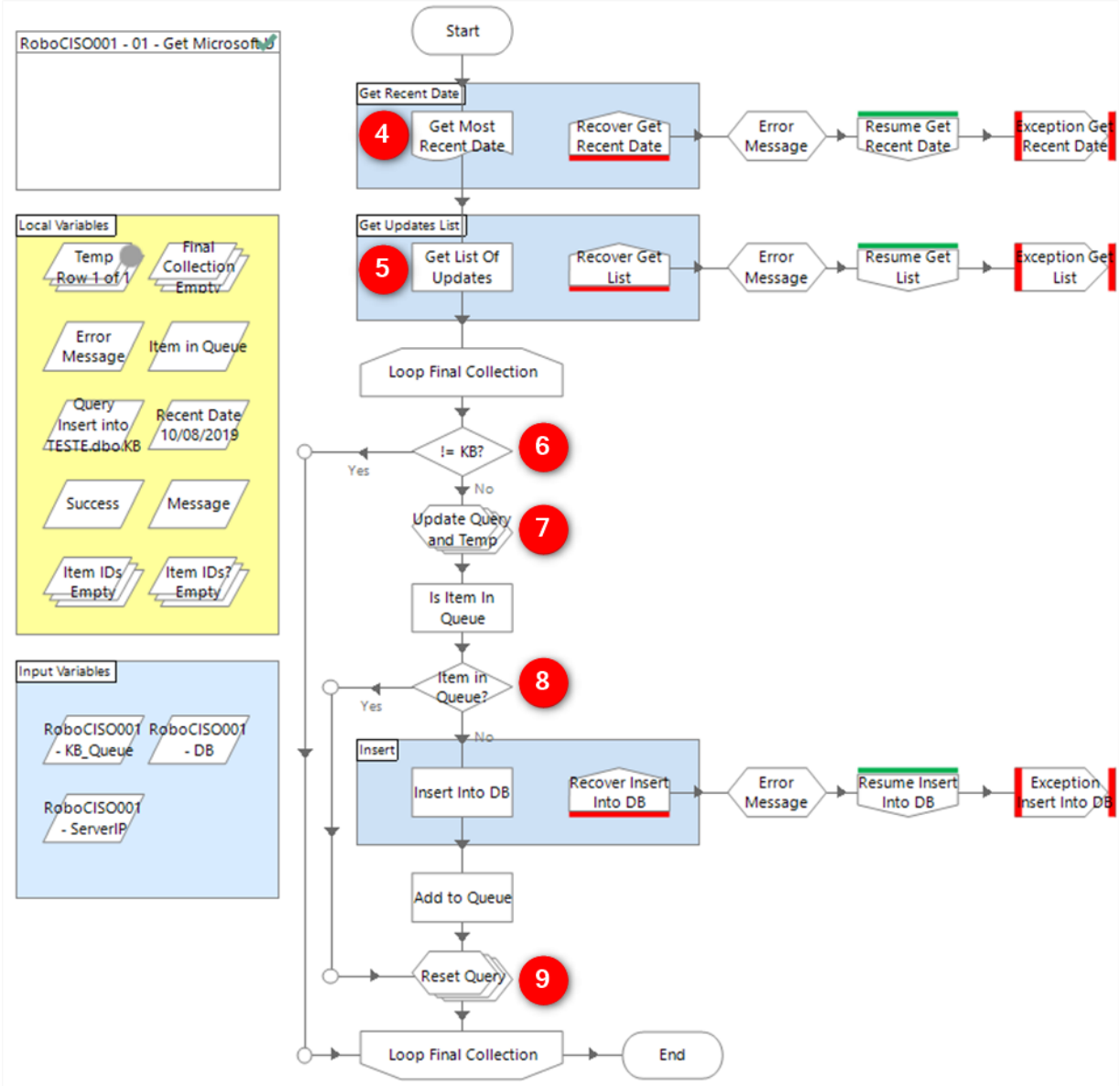
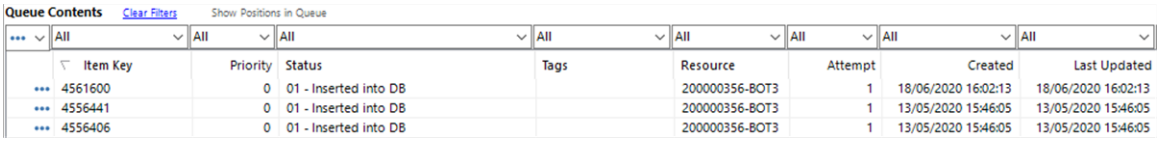



Figura 4.2: Página *Get Updates List*.

A *query* utilizada para a inserção da informação na base de dados é composta por quatro campos: ***KB_Date***, ***Article***, ***Type*** ***Applies_To*** descritos na Subseção 3.4.1.

```
INSERT INTO KB_list
(KB_Date, Article, Type, Applies_To) VALUES
('DATEFIELD','ARTICLEFIELD','TYPEFIELD','APPLIESTOFIELD')
```




| Item Key | Priority | Status | Tags | Resource | Attempt | Created | Last Updated |
|---|---|---|---|---|---|---|---|
| 4561600 | 0 | 01 - Inserted into DB | | 200000356-BOT3 | 1 | 18/06/2020 16:02:13 | 18/06/2020 16:02:13 |
| 4556441 | 0 | 01 - Inserted into DB | | 200000356-BOT3 | 1 | 13/05/2020 15:46:05 | 13/05/2020 15:46:05 |
| 4556406 | 0 | 01 - Inserted into DB | | 200000356-BOT3 | 1 | 13/05/2020 15:46:05 | 13/05/2020 15:46:05 |

Figura 4.3: Excerto da fila de trabalho *RoboCISO001 - KBs*.

Após as validações, o conteúdo da coleção temporária é eliminado e a *query* volta a ficar no seu estado inicial, ***Reset Query*** (9). Quando a iteração da *Final Collection* termina, o processo volta à página inicial ilustrada na Figura 4.1 onde é executado o ***Terminate*** (3), fechando o Chrome deixando o ambiente no seu estado inicial.

### 4.2.2 RoboCISO001 - 02 - Get CVE list from Outlook

No processo *RoboCISO001 - 02 - Get CVE list from Outlook* são obtidas as vulnerabilidades divulgadas mensalmente pela Microsoft e que são enviadas via *e-mail* para o RoboCISO. A aplicação utilizada por este processo é o Outlook. A implementação deste processo, cuja *Main Page* está ilustrada na Figura 4.4, está divida em três partes principais: ***Get email*** (1), ***Open Email and get Table*** (3) e ***Insert into DB and Queue*** (4). A primeira ação, ***Get email*** (1), verifica se existem novos *e-mails* com o assunto *"Patch Tuesday"*. Para tal é utilizado um VBO disponibilizado pelo BP, *Get Received Items (Basic)* que recebe como parâmetro o *Subject* e retorna uma coleção com o *Id* do *e-mail*, caso este exista. Se a coleção estiver vazia, significa que não existe nenhum novo *e-mail*, caso contrário, existe e será guardado na máquina num ficheiro de extensão *.msg*. De volta à *Main Page*, é verificado se a coleção *Items* está preenchida, ***Items Empty?*** (2):

- Se estiver vazia, a execução do processo termina;
- Se contém informação, passa à fase seguinte contida da página ***Open Email and get Table*** (3), onde é feito o processamento do conteúdo do *e-mail*. O *e-mail*, que contém uma tabela com as diferentes vulnerabilidades, é aberto recorrendo ao terminal e os elementos da tabela são obtidos e inseridos numa coleção, *CVEs Table*, com o auxílio da ação criada para esse efeito, ***Get Table***, do objeto *RoboCISO001 - Outlook Interaction*.

A execução retorna à *Main Page*, o terminal e o *e-mail* são fechados, com as ações ***Close cmd*** e ***Close Email Outlook***, respetivamente. De seguida a coleção *CVEs Table* é iterada e os seus itens são inseridos na base de dados e na fila de trabalho *RoboCISO001 - CVE*, dedicada às vulnerabilidades, através da página ***Insert into DB and Queue*** (4) com os seguintes parâmetros:

- *Item Key*, corresponde ao identificador único da vulnerabilidade, por exemplo: ***CVE-2020-1173***;
- *Status*, após ser inserido na base de dados está definido como ***'01 - Inserted into DB'***;
- *Tag*, possui a severidade da vulnerabilidade que pode ter um de quatro valores possíveis: ***Critical, Important, Moderate*** ou ***Low***.

Este passo é muito semelhante ao explicado na Subseção 4.2.1 e ilustrado na figura Figura 4.2. A *query* utilizada é composta por seis campos, explicados na Subseção 3.4.2:

```
INSERT INTO CVE_list
(CVE_ID, Vulnerability_Type, Products,
        Severity, Workaround, Exploited)
VALUES ('CVEID','VULNTYPES','PRODUCTS',
        'SEVERITY','WORKAROUND','EXPLOITED')
```

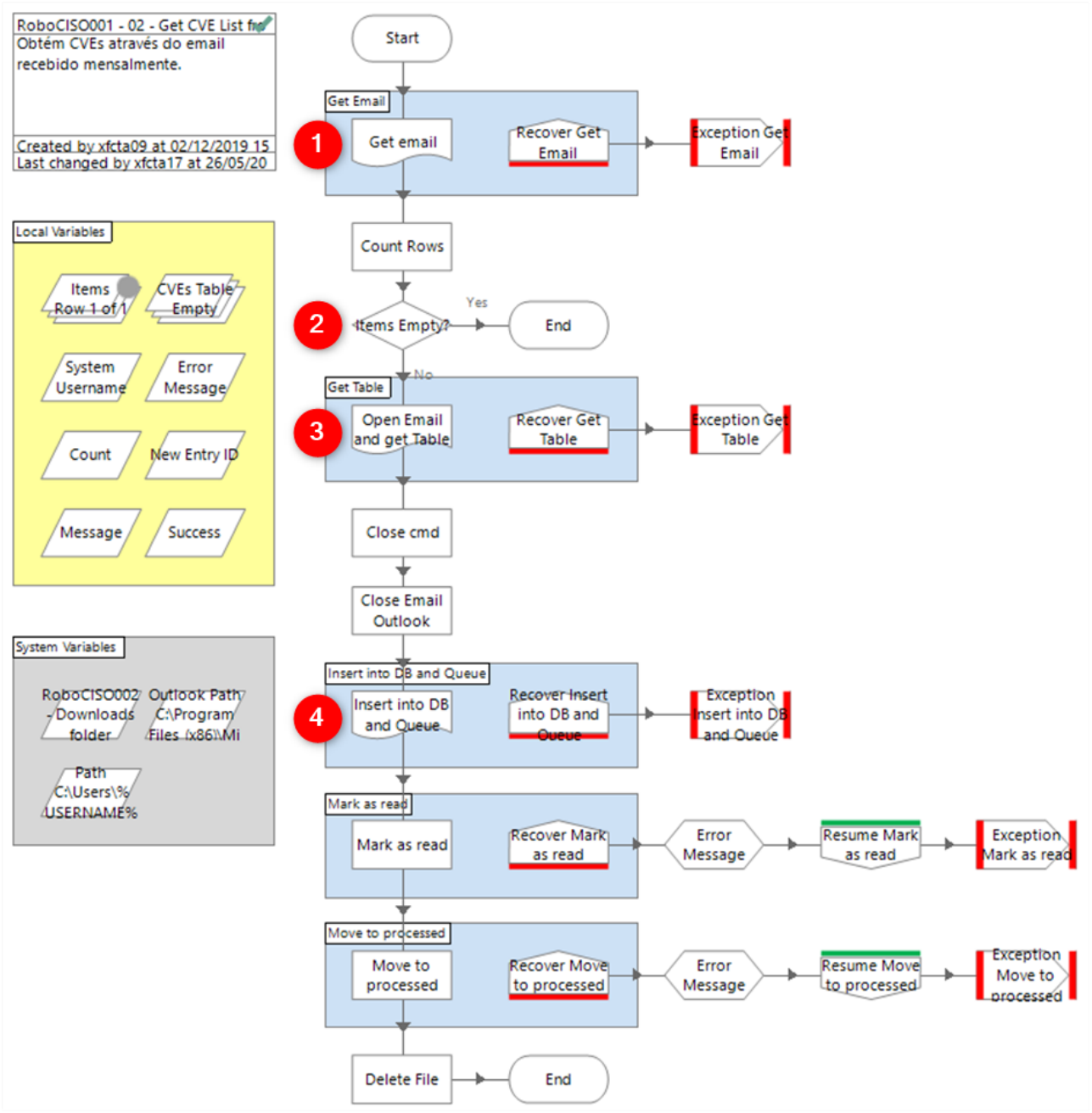



Figura 4.4: *Main Page* do processo *RoboCISO001 - 02 - Get CVE_list from Outlook*.

### 4.2.3 RoboCISO001 - 03 - Get KBs and Severity from each CVE

Este processo é responsável por obter, através do *site* da Microsoft, para cada vulnerabilidade identificada no processo anterior, **se esta for crítica**, a lista de *patches* que a mitigam. A *Main Page* deste processo está ilustrada na Figura 4.5, estando divida em três partes: ***Launch*** (1), ***Get CVEs From Queue*** (2) e ***Terminate*** (3). Os passos assinalados com (1) e (2) são idênticos aos descritos na Subseção 4.2.1 para o processo *RoboCISO001 - 01 - Get Microsoft Updates List*, sendo a única alteração o *url* utilizado no *launch*, neste caso que será o *url* da vulnerabilidade iterada.

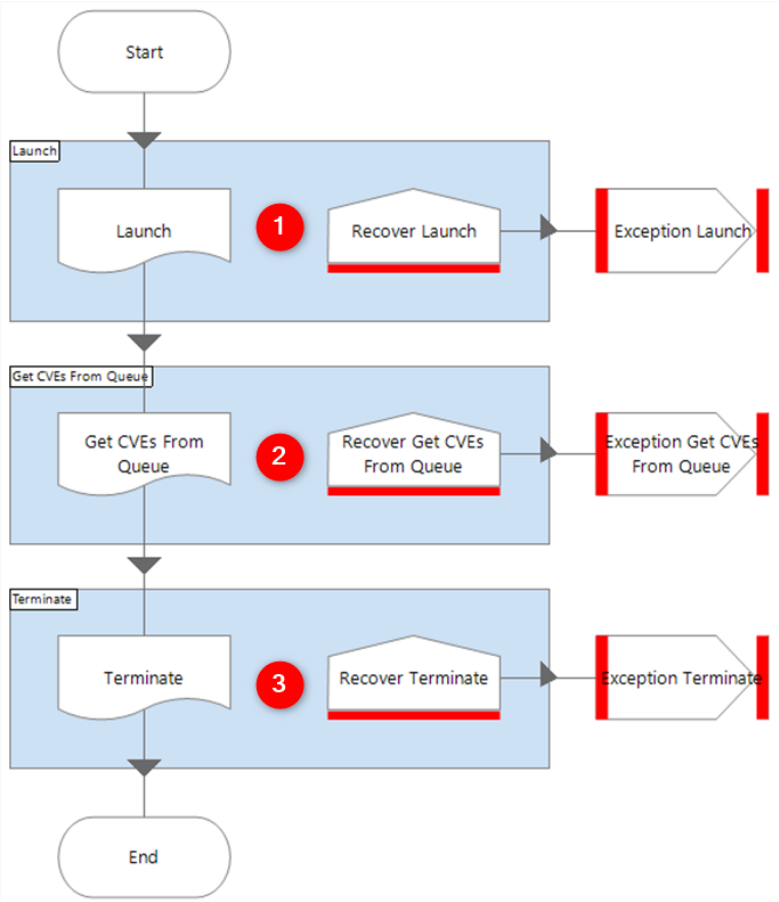



Figura 4.5: *Main Page* do processo *RoboCISO001 - 03 - Get KBs and Severity from each CVE*.

Uma vez que grande parte do processo se desenrola na página ***Get CVEs From Queue*** (2), esta está em detalhe na Figura 4.6. A primeira ação executada é um ***Set Connection*** para estabelecer a ligação à base de dados. De seguida é executada a ação ***Get Next Item*** (4), que obtém da fila de trabalho *RoboCISO001 - CVE* uma das vulnerabilidades pendentes, devolvendo o seu *ItemID*.

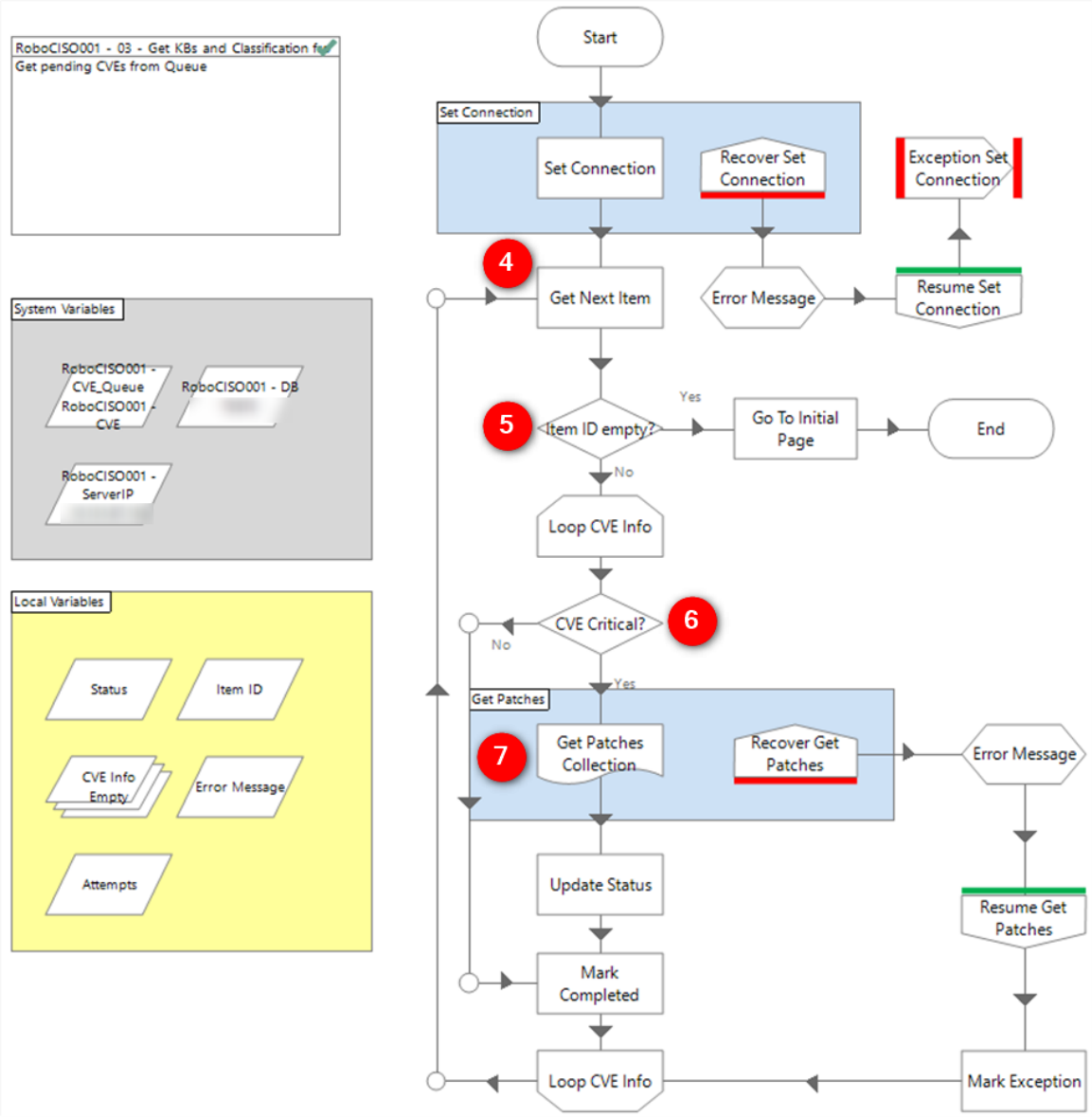



Figura 4.6: Etapa de seleção das vulnerabilidades a serem pesquisadas.

Se o *ItemID* estiver vazio, ***Item ID empty?*** (5), significa que não existem itens pendentes e a execução volta para a *Main Page*, sendo feito o ***Terminate*** do Chrome (3) e a execução termina, caso contrário, itera sobre a coleção resultante, *CVE Info* e verifica se a vulnerabilidade tem severidade *Critical*, ***CVE Critical?*** (6):

- Se não, marca o elemento como *Completed* e passa ao próximo item da fila de trabalho;
- Se sim, irá obter os *patches* que a mitigam a partir da página, ***Get Patches Collection*** (7), em detalhe na Figura 4.7. Após fazer o *Attach*, utiliza o objeto que interage com a página *web* da vulnerabilidade e obtém todos os *patches*, ***Get Patches for CVE*** (8). Por fim, insere a nova informação na base de dados e na fila de trabalho *RoboCISO001 - CVE_KB*, ***Insert Into DB and Queue*** (9) com os parâmetros:
  - *Item Key*, corresponde aos identificadores únicos da vulnerabilidade e do *patch*, ficando na

forma ***CVE - KB*** por exemplo: *CVE-2020-1173 - 4556799*;

- *Status*, definido como ***'01 - Inserted into DB'***;
- *Tag*, possui a severidade que a Microsoft associou ao *patch* podendo ter um de quatro valores possíveis: ***Critical, Important, Moderate*** ou ***Low***;
- Uma vez que não serão executadas mais tarefas com estes itens, são marcados como *Completed*;

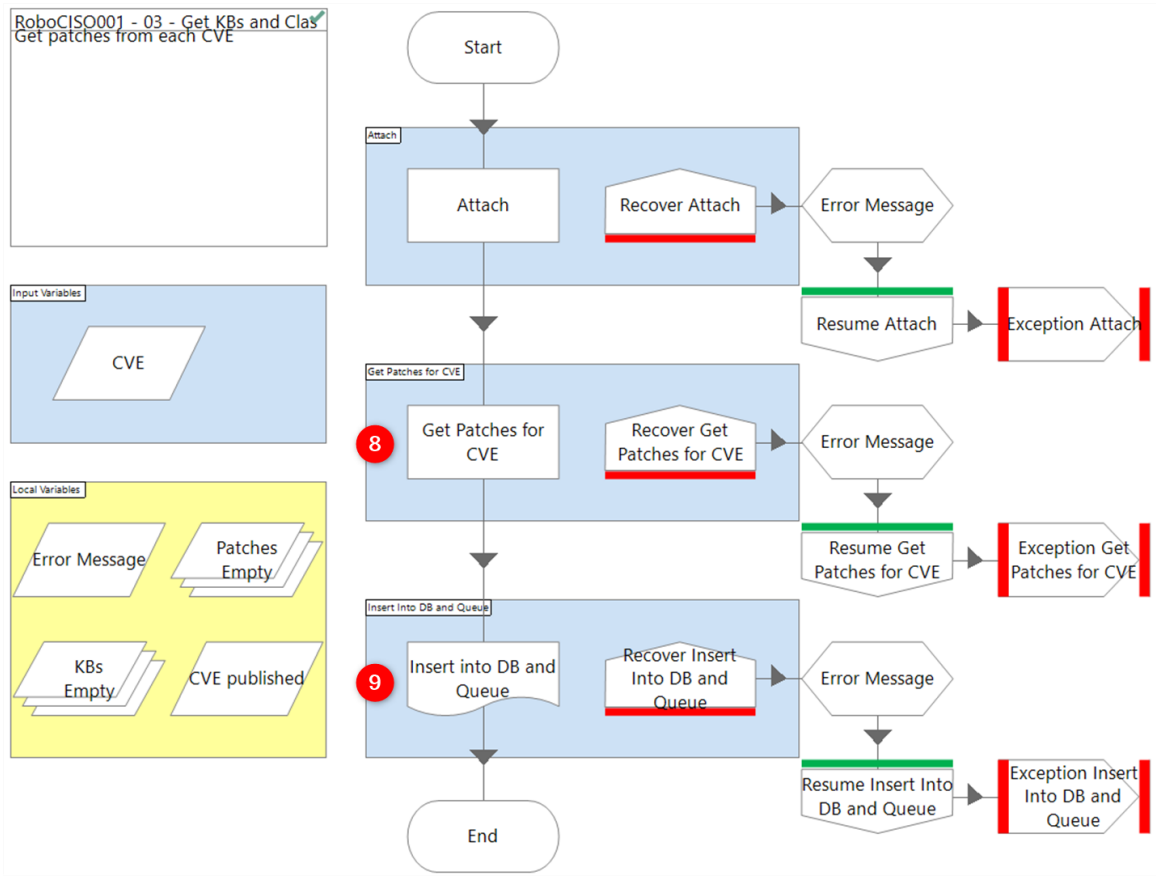



Figura 4.7: Etapa de obtenção dos *patches* para cada vulnerabilidade.

A *query* utilizada na inserção da nova informação na base de dados é a seguinte:

```
INSERT INTO CVE_KB
        (CVE, CVE_published, Product, Platform,
        Article, Impact, Severity, Supersedence)
VALUES ('CVEFIELD','CVEPUBLISHEDFIELD','PRODUCTFIELD',
        'PLATFORMFIELD', 'ARTICLEFIELD', 'IMPACTFIELD',
        'SEVERITYFIELD', 'SUPERSEDENCEFIELD')
```

Voltando à página ***Get CVEs From Queue***, ilustrada na Figura 4.6, o *status* na fila de trabalho da vulnerabilidade é atualizado para *'02 - KBs Obtained'*, através da ação ***Update Status*** e marca o item como *Completed* através da ação ***Mark Completed***.

### 4.2.4 RoboCISO001 - 04 - Classify KBs

Neste processo é chamada uma *Stored Procedure*, que corresponde a código SQL[4] que está guardado e pronto a usar sempre que necessário, para classificar os *patches*. A cada *patch* é atribuído um nível de severidade que corresponde ao valor máximo encontrado na informação recolhida no processo anterior. O processo está ilustrado na Figura 4.8.

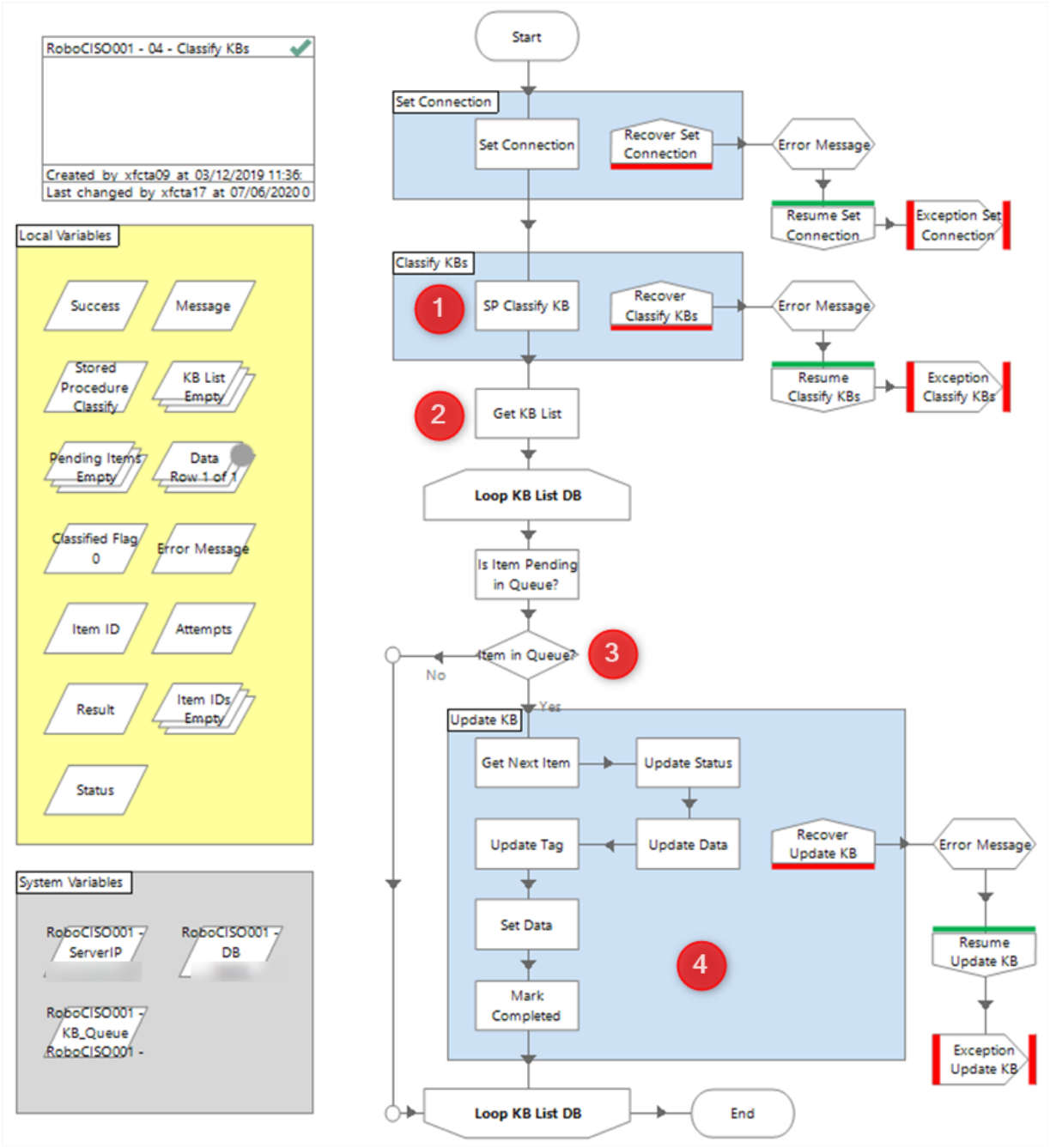



Figura 4.8: Etapa de classificação dos *patches*.

A execução do processo inicia com a ligação à base de dados e de seguida a execução da *stored procedure*, ***SP Classify KB*** (1), que consiste na *query* representada no Apêndice A.

[4]Structured Query Language, é a linguagem padrão utilizada em bases de dados relacionais

Após a classificação dos *patches* é obtida a coleção de *patches* classificados através da ação ***Get KB List*** (2). Essa coleção é iterada de seguida e verifica-se se o *patch* iterado está pendente na fila de trabalho *RoboCISO001 - KB List* através da ação ***Is Item Pending in Queue?***, retornando a *flag* com valor *True*, se está pendente ou *False*, se não está pendente. Na decisão ***Item in Queue?*** (3) o seu valor é lido:

- Se a *flag* estiver a *True*, as ações dentro do bloco azul (4), são executadas:
  1. ***Get Next Item***, obtém a coleção com a informação do item;
  2. ***Update Status***, atualiza o *Status* do item de ***'01 - Inserted into DB'*** para ***'02 - KB Classified'***;
  3. ***Update Data***, atualiza o campo *Severity* com a classificação agora existente (antes este campo estava vazio);
  4. ***Update Tag***, atualiza a *Tag* para o nível de severidade associado ao *patch*;
  5. ***Set Data***, a coleção é atualizada na fila de trabalho;
  6. ***Mark Completed***, o item é marcado como *Completed*.
- Se a *flag* estiver a *False*, passa ao próximo item;

Quando termina a iteração da coleção *KB List*, termina a execução.

## 4.3 RoboCISO002 - Alerts

O RoboCISO002 - Alerts é composto por cinco processos independentes cuja implementação se encontra descrita nas próximas secções.

### 4.3.1 RoboCISO002 - Patching Alert

O processo *RoboCISO002 - Patching Alert* é responsável pela deteção e envio das notificações de *patching*. A sua *Main Page* está ilustrada na Figura 4.9. A execução começa com a obtenção do *Id* da nova notificação, ***Get Last Alert ID PA*** (1). Este *Id* terá a estrutura *PA#000000*. De seguida é executada a *query* de deteção de *patches* em atraso, ***Get Critical Patching List*** (2) estando esta representada nas secção A.2 do Apêndice A. Se a *query* retornar resultados a coleção *Critical Patching* será preenchida com os mesmos. Serão adicionados alguns campos extra à coleção *Critical Patching* que serão preenchidos na página ***Get Additional Info CP*** (3). Estes campos adicionais estão descritos na Tabela 4.1 e estarão presentes no *e-mail*, mas não na mensagem WhatsApp de forma a mantê-la simples e curta.

| Campos | Descrição |
|---|---|
| *Type* | Tipo de produtos aos quais o *patch* se aplica |
| *Applies To* | Sistemas operativos aos quais o *patch* se aplica |
| *Link* | Link para a página do *patch* da Microsoft |
| *Impact* | Tipo de impacto que a não instalação pode ter |
| *Criteria* | Qual dos critérios foi satisfeito |

Tabela 4.1: Campos adicionais enviados no e-mail do *Patching Alert*.

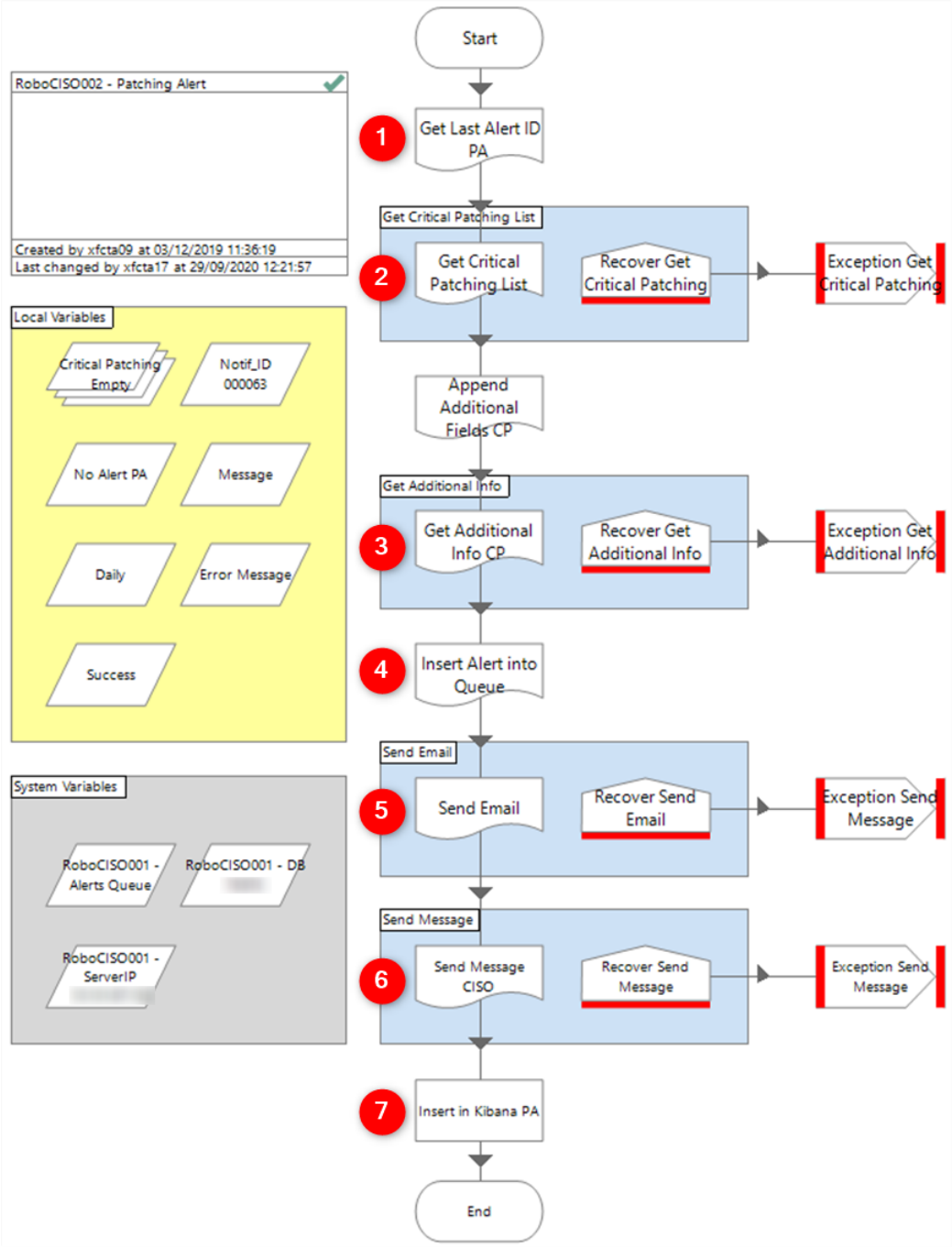



Figura 4.9: *Main Page* do processo *Patching Alert*.

De seguida a coleção é inserida na fila de trabalho *RoboCISO002 - Alerts*, através da página ***Insert Alert into Queue*** (4), com os seguintes parâmetros:

- O *Status* fica com o valor de ***'01 - Alert Detected'***;
- A *Tag* fica com o valor ***'Patching Alert#Date'***, por exemplo: *Patching Alert#2020.04.12*;
- Se a coleção estiver vazia, significa que não existem *patches* que tenham atingido as condições para ser gerada a notificação, sendo colocada uma entrada com a *Item Key* ***Patching#No alert***, caso contrário, para cada elemento da coleção será inserida uma entrada com a *Item Key* ***Patching#KB***,

onde o KB corresponde ao *Id* na *Knowledge Base* da Microsoft e que funciona como o identificador único deste *patch*.

De seguida o *e-mail* é enviado para o seu conjunto de destinatários a partir da página ***Send Email*** (5), em detalhe na Secção B.2. Foram criadas duas variáveis de ambiente[5] com os destinatários do *e-mail*:

- *RoboCISO001 - Alert To*, que contém o endereço de *e-mail* do CISO;
- *RoboCISO001 - Alert Cc*, que contém os endereços de *e-mail* dos restantes membros da equipa da DCY que o CISO pretende que recebam a notificação;

Após o envio do *e-mail* o *status* dos itens na fila de trabalho é alterado para ***'02 - Email Sent'***. O próximo passo é o envio das notificação via WhatsApp para o grupo RoboCISO previamente criado, através da página ***Send Message CISO*** (6). Esta página encontra-se em detalhe na Secção B.3 onde se pode observar que, caso a coleção final possua mais do que três elementos, apenas os três primeiros serão enviados na mensagem WhatsApp, de forma a que esta não fique demasiado longa. Estes e os restantes itens serão todos enviados no respetivo *e-mail*, juntamente com as informações adicionais obtidas no passo (3). Após o envio da mensagem o *status* dos itens na fila de trabalho é atualizado para ***'03 - WhatsApp Sent'*** e são marcados como *Completed*. Na Secção C.2 estão ilustrados exemplos dos *e-mails* e mensagens WhatsApp gerados por este processo. Por fim, a coleção que contêm os itens enviados na notificação e *e-mail* é inserida no HIDRA, ***Insert in Kibana PA*** (7).

### 4.3.2 RoboCISO002 - DDoS Alert

Este processo é responsável por detetar e gerar alertas em caso de ataques DoS/DDoS, que tenham como alvo clientes ou o próprio grupo Altice Portugal e que ultrapassem as condições de alerta especificadas na Subseção 3.5.2. A *Main Page* deste processo está ilustrada na Figura 4.10.

O primeiro passo na execução deste processo é ***Get Info From DB*** (1), onde é estabelecida a ligação à base de dados do RTIR de onde são obtidos os *tickets* referentes a ataques de Dos/DDoS. A *query* utilizada para obter os *tickets* está representada na Secção A.3 do Apêndice A. É retornada uma coleção com a informação presente na base de dados do *RTIR*, contendo os campos descritos na Tabela 3.6, cujos elementos serão processados na página ***Get Content of Arbor*** (2). Nessa página é calculada a duração do ataque fazendo a diferença entre os campos *Resolved_Dt* e *Created_Dt*. Caso o *ticket* ainda esteja aberto, será considerado como *Resolved_Dt* o dia e hora local. O campo *MailBody* é igualmente processado uma vez que é aqui que está a informação do volume do ataque, sendo necessário recorrer a uma expressão regular para o obter.

Uma vez que podem ser gerados vários *tickets* para um mesmo ataque, apenas aquele que apresenta maior volume é guardado eliminando os duplicados e os de menor volume através da página ***Remove Arbor Duplicates***. De seguida os elementos da coleção *Arbor* são filtrados, na página ***Filter DDoS Attacks*** (3), e todos aqueles que não satisfizerem as condições de alerta são eliminados da coleção. Quando a filtragem termina a execução volta à *Main Page* onde é obtido o *Id* do novo alerta a ser gerado que terá a estrutura *ARB#000000*.

[5] Variável BP cujo valor é fixo e não alterável através do processo

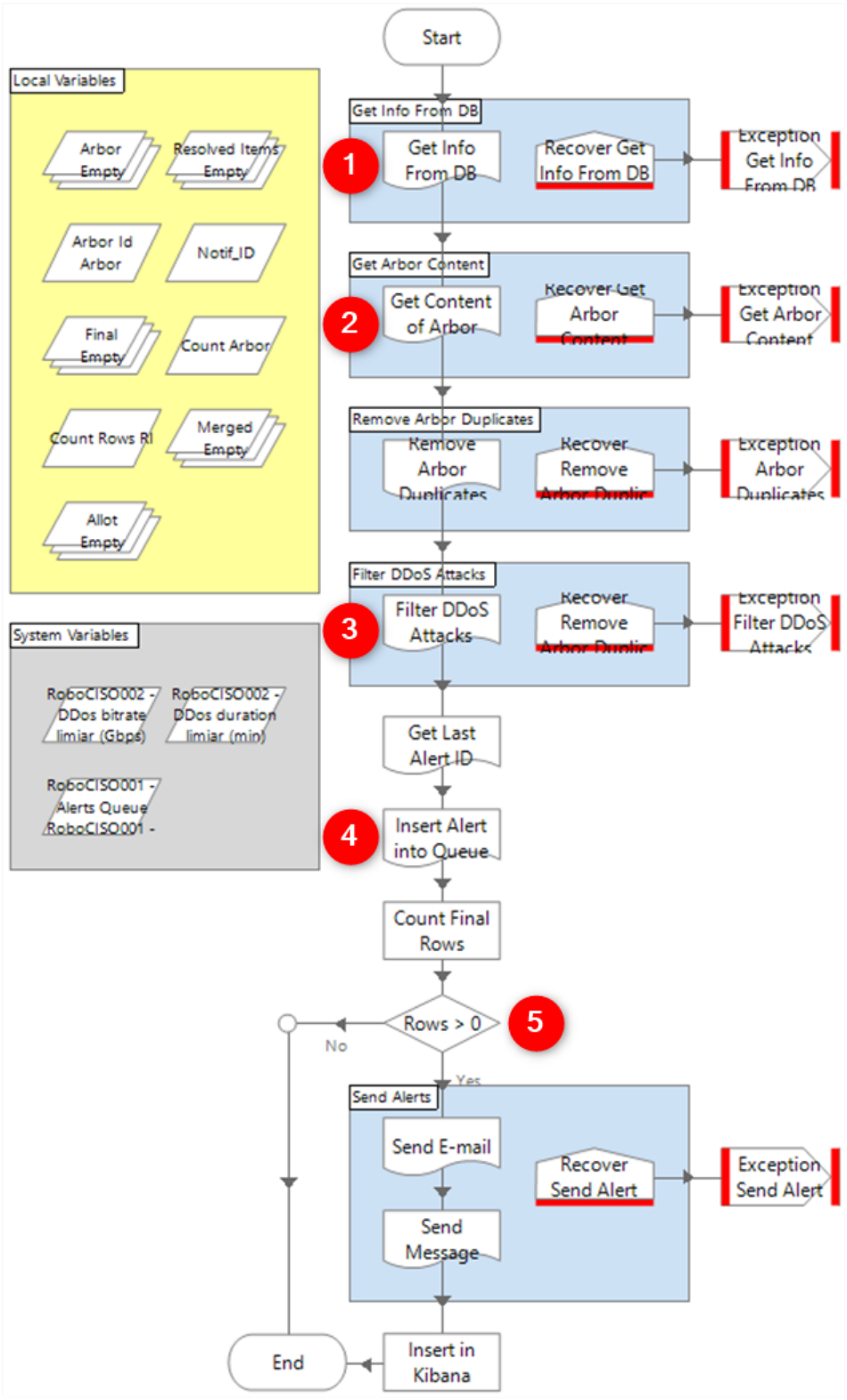



Figura 4.10: *Main Page* do processo *DDoS Alert*.

A execução passa para a página ***Insert Alert into Queue*** (4) onde a coleção *Arbor* é iterada e para cada elemento é verificado:

- Se o seu estado é *open*, é inserido na fila de trabalho e colocada na coleção *Final*, que irá conter os elementos a reportar;

- Se o seu estado é *resolved*, verifica-se a última entrada na fila de trabalho para este elemento está marcada a *open*:
  - Se sim, é inserida uma nova entrada agora com a tag *resolved* e o elemento é adicionado à coleção *Final*;
  - Se não, item não será adicionado de novo à fila nem à coleção de elementos a reportar.

Este passo garante que enquanto o incidente não for marcado como resolvido será enviado um alerta e, que ao ser resolvido essa informação será reportada uma única vez. A *Item Key* de cada incidente a reportar terá o formato ***Arbor#Id#RTIR ID***, por exemplo *Arbor#21#123456* e o seu *status* ***'01 - Alert Detected'***. Se não existirem elementos na coleção *Final*, é inserida uma entrada na fila de trabalho *RoboCISO002 - Alerts*, cuja *Item Key* terá a estrutura ***RoboCISO#yyyy.MM.dd HH:mm:ss***, onde, *yyyy.MM.dd HH:mm:ss* corresponde à data e hora em que o elemento foi processado e o *status* será ***'01 - No DDoS Alerts'***. Após este passo a execução retorna à *Main Page* a coleção *Final* que contém os elementos a reportar verificando se esta lista está vazia no passo ***Rows>0*** (5). Se estiver vazia a execução termina, caso contrário o *e-mail* será enviado, através da página ***Send E-mail***, para o mesmo conjunto de *e-mails* do processo anterior, contido na variável *RoboCISO001 - Alert To* e o *status* na fila de trabalho dos itens reportados será atualizado para ***'02 - Email Sent'***. De seguida é enviada a mensagem via WhatsApp, através da página ***Send Message***, para o grupo RoboCISO e o *status* novamente atualizado para ***'03 - WhatsApp Sent'*** sendo finalmente marcados como *Completed*. Na Secção C.3 estão ilustrados exemplos dos *e-mails* e mensagens WhatsApp gerados por este processo. O último passo da execução é a inserção da coleção *Final* no HIDRA, através da ação ***Insert in Kibana***.

### 4.3.3 RoboCISO002 - SLA Exceeded

Este processo é responsável por detetar e gerar notificações em caso de incumprimento dos SLAs de tempo máximo de resolução de *tickets* do RTIR, referentes a incidentes internos à Altice Portugal. A *Main Page* deste processo está ilustrada na Figura 4.11 e tal como acontece nos restantes processos já descritos, obtém-se o *Id* da notificação, neste caso a partir da página ***Get Last Alert ID***, tendo este a estrutura *SLA#000000*. De seguida é obtida a informação referente aos *tickets* através da base de dados, na página ***Get Info From DB*** (1), onde é executada a *query* representada na Secção A.4 do Apêndice A. O resultado da *query* é armazenado na coleção *RTIR* que será processada na página ***Calc SLA Exceeded*** (2). Nesta página é calculado o número de horas em que o SLA foi excedido e o número de horas passadas até à resolução do *ticket*, ou caso ainda esteja aberto, o número de horas até ao momento. A coleção *RTIR* é retornada de volta para a *Main Page*, já tratada e com o campo extra da duração, sendo feita uma contagem do seu número de linhas, ***Count Rows Before***.

Se a coleção estiver vazia a execução termina, caso contrário, a execução continua na página ***Insert Alert into Queue*** (3). A coleção *RTIR* será iterada e o seu conteúdo adicionado à fila de trabalho *RoboCISO002 - Alerts*. Antes de adicionar o elemento iterado à fila de trabalho, é verificado se:

- O estado do *ticket* é *open*, é inserido na fila de trabalho com a *Item Key* ***SLA#Id#RTIR ID***, por exemplo *SLA#40#123456*, com o *status* ***'01 - Alert Detected'*** e *tag* ***open***;

- O estado do *ticket* é *resolved*, verifica se a última entrada na fila de trabalho associada a este *ticket* tem a *Tag* marcada a *open*:
  - Se sim, é inserida uma nova entrada com a *Tag* ***resolved***;
  - Se não, o item não será colocado novamente na fila e será removido da coleção *RTIR*.

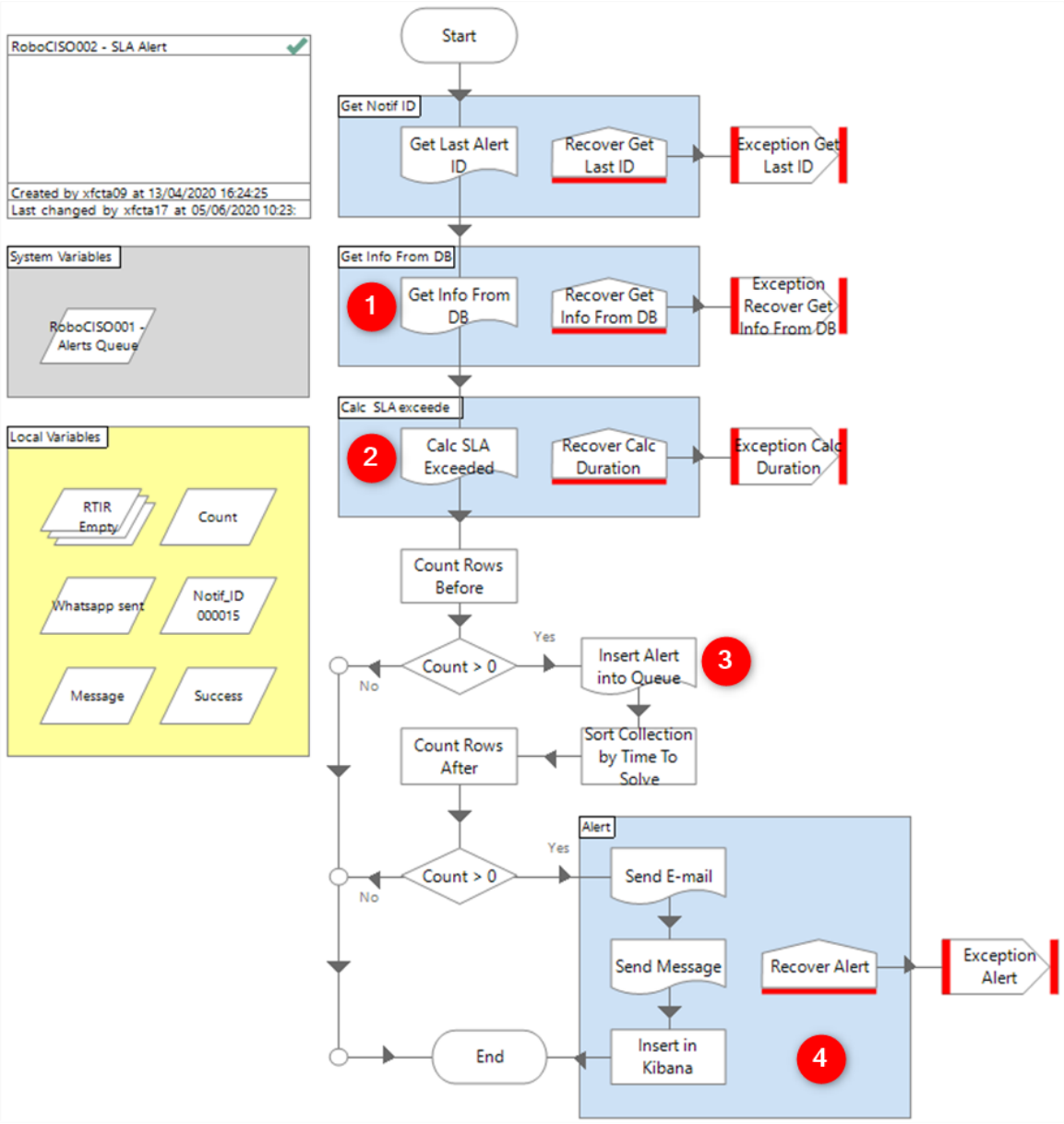



Figura 4.11: *Main Page* do processo *SLA Exceeded*.

Após a iteração da coleção, a execução volta para a *Main Page*, onde é novamente verificado se a coleção está vazia através da ação ***Count Rows After***, devido à filtragem de casos *resolved* realizada na página anterior. Se estiver vazia a execução termina, caso contrário, os passos representados no bloco azul (4) serão executados. O *e-mail* será enviado através da página ***Send E-mail*** e o *status* dos itens enviados passa a ***'02 - Email Sent'***. De seguida, será enviada a mensagem via WhatsApp e o *status* dos elementos será novamente atualizado para ***'03 - WhatsApp Sent'*** através da página ***Send Message***. Na

Secção C.4 estão ilustrados exemplos dos *e-mails* e mensagens WhatsApp gerados por este processo. Por fim, a coleção será inserida no HIDRA, ***Insert in Kibana***.

### 4.3.4 RoboCISO002 - Bitsight Alert

Este processo é responsável pela deteção e geração de alertas caso o *Bitsight rating* de alguma empresa do grupo Altice sofra alterações superiores a 5%, sejam estas positivas ou negativas. O processo implementado encontra-se ilustrado nas Figuras 4.12 e 4.13. Para poder implementar este processo foi necessário criar uma conta para o RoboCISO na plataforma da Bitsight. Esta conta permite definir os parâmetros para receção de alertas via *e-mail*. Ficou definido que sempre que existir uma alteração, deverá ser enviado um *e-mail* para o RoboCISO.

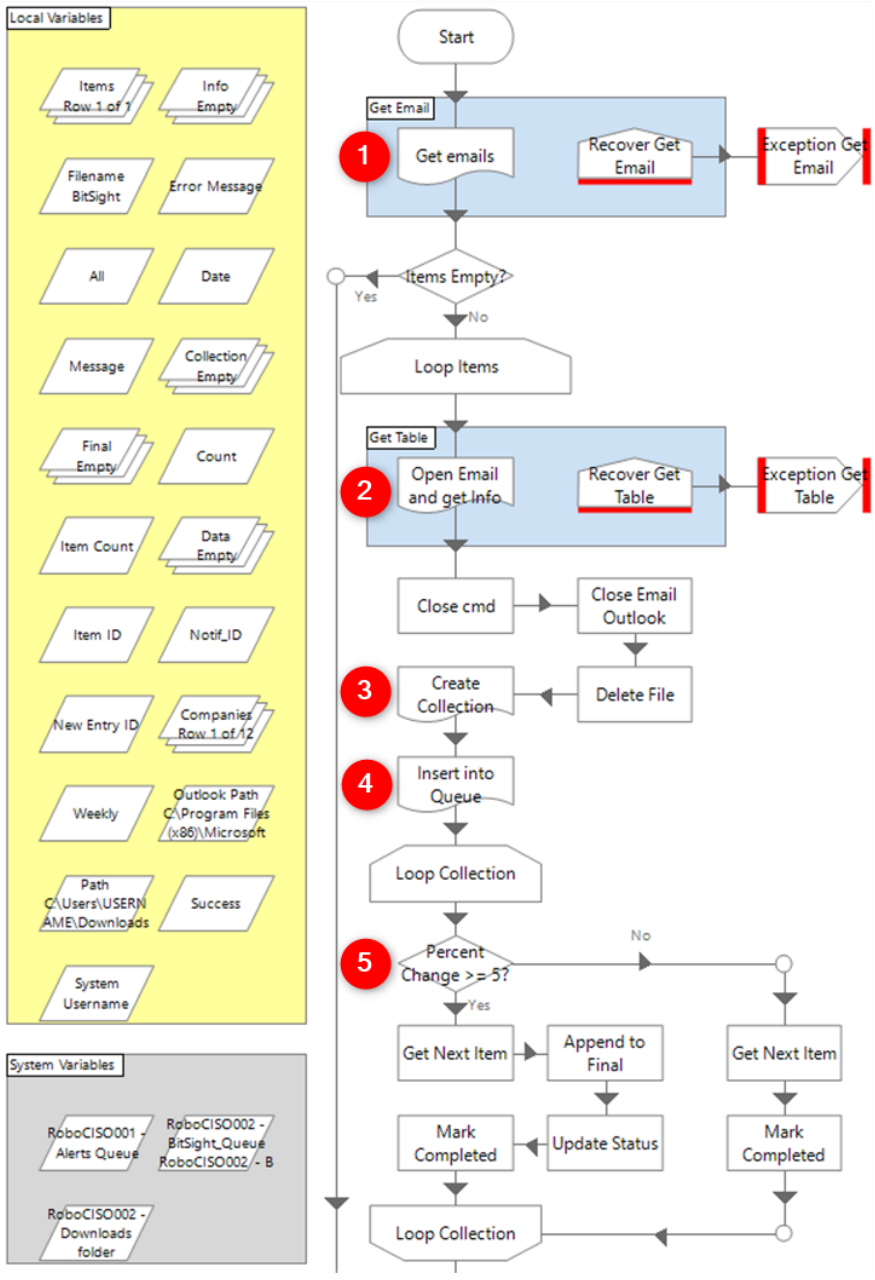



Figura 4.12: Primeira parte da *Main Page* do processo *Bitsight Alert*.

A primeira etapa deste processo, consiste em verificar se existem novos *e-mails* enviados pela Bitsight, na caixa de entrada do *e-mail* do RoboCISO, ***Get emails*** (1). Se existirem, significa que houve uma alteração em um ou mais *ratings*, e os *Id*s desses *e-mails* serão retornados numa coleção com o nome de *Items*. Para além de produzir essa coleção os *e-mails* existentes são transferidos para a máquina num ficheiro com extensão *.msg*.

Se a coleção possuir elementos, estes serão iterados:

1. Na página ***Open Email and get Info*** (2) cada *e-mail* é aberto e o seu conteúdo processado, retornando duas variáveis: *All* que contém o conteúdo textual do *e-mail* e *Date*, que contém a data do alerta Bitsight;

2. Uma vez que cada *e-mail* é aberto através do terminal, passo que está contido dentro da página ***Open Email and get Info*** (2), o terminal é fechado ***Close cmd***, o *e-mail* é fechado ***Close Email Outlook*** e o ficheiro é apagado da máquina através da ação ***Delete File***;

3. De seguida é criada uma coleção com o nome *Collection*, através da página ***Create Collection*** (3), onde o conteúdo das variáveis *All* e *Date* é processado, contendo os campos descritos na Tabela 4.2;

4. A coleção é inserida na fila de trabalho *RoboCISO002 - Bitsight Alerts*, ***Insert into Queue*** (4) com a *Item Key* de formato; ***'Company AlertDate'***, por exemplo: *CyW-07z-CORP 2020-06-13*, o *status* ***'01 - Alert Processed'*** e a *Tag* será ***rating_antes*** $\rightarrow$ ***rating_depois (%)*** *e.g., 650* $\rightarrow$ *660 (1%)*;

5. A *Collection* é iterada verificando se a alteração do *rating* é superior a 5% ou inferior a -5%, ***Percent Change >= 5*** (5):

    - Se sim, esse item é recolhido da fila de trabalho *RoboCISO002 - Bitsight*, ***Get Next Item*** e adicionado a uma coleção com o nome *Final*. O *status* do elemento é atualizado para ***'02 - Alert Detected'*** e é marcado como *Completed*;
    - Se não, o item é apenas marcado como *Completed*.

| Campos | Descrição |
|---|---|
| *Company* | Código da empresa |
| *Date* | Data da alteração do *rating* |
| *Change* | *Rating* antigo, *rating* atual e percentagem da alteração |
| *Reason* | Em caso de descida, a sua razão |

Tabela 4.2: Campos da tabela enviada no e-mail *BitSight Alert*.

Após a iteração da coleção *Collection* a primeira parte da *Main Page* está concluída. O próximo passo é obter o *Id* do novo alerta a ser gerado, através da página ***Get Last Alert ID*** (6) que terá a estrutura *BS#000000*. Se a coleção *Items* estiver vazia, verificado em ***Items Empty?***, após a execução da página ***Get Emails*** (1), a execução passa imediatamente para a página ***Insert Alert into Queue*** (7).

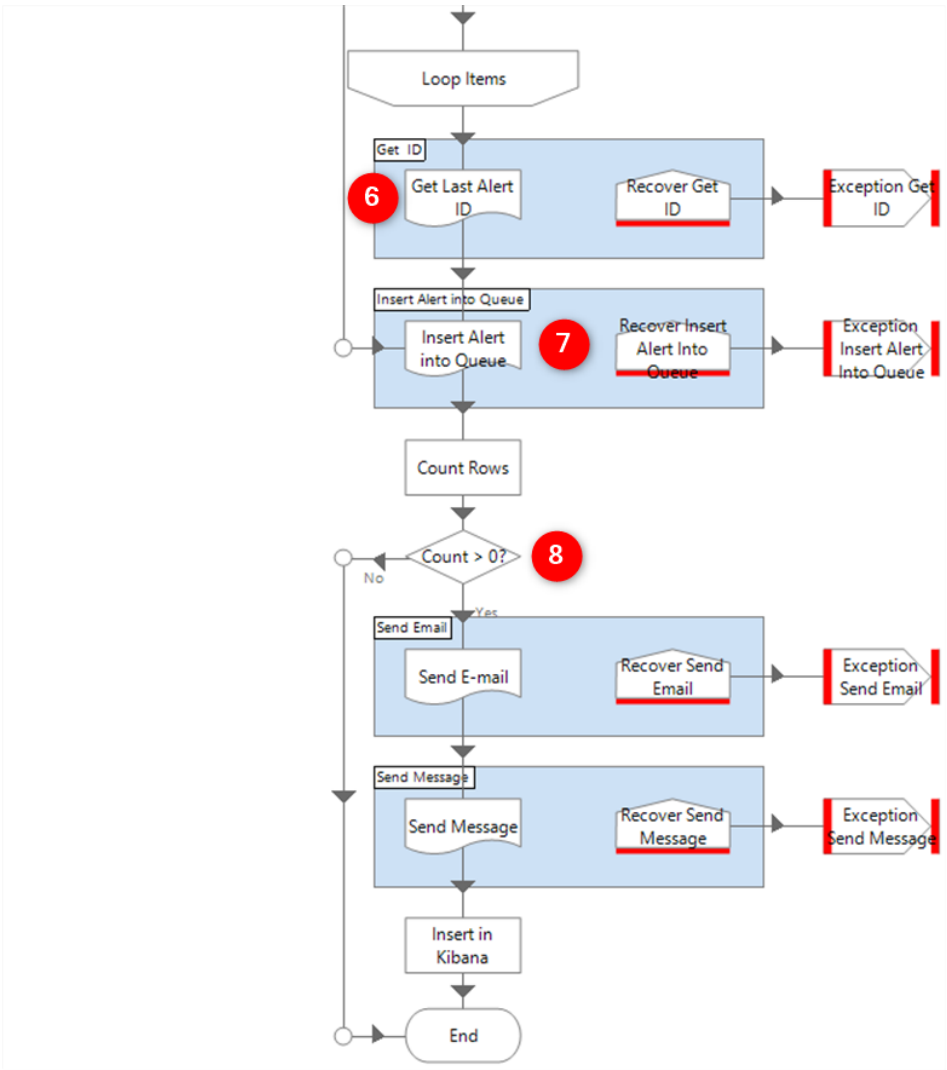



Figura 4.13: Segunda parte da *Main Page* do processo *Bitsight Alert*.

O conteúdo da coleção *Final* é inserido na fila de trabalho *RoboCISO002 - Alerts* com os mesmos parâmetros utilizados na inserção dos itens da fila de trabalho *RoboCISO001 - Bitsight Alerts*, descritos no passo 4. Se não existirem alertas é colocada uma entrada na fila de trabalho *RoboCISO002 - Alerts*, com uma *Item Key* de estrutura ***RoboCISO#yyyy.MM.dd HH:mm:ss***, tal como acontece no *DDoS Alert*, e com o *status* ***'01 - No Bitsight Alerts'***. A execução retorna à *Main Page* onde é feita uma contagem do número de linhas da coleção *Final* no passo ***Count >0?*** (8):

- Se não for maior que zero, não existem alertas e a execução termina;
- Se for maior que zero, significa que existem alertas e portanto, o *e-mail* e a mensagem são gerados e enviados, tal como acontece nos restantes processos, utilizando os mesmos *status* anteriormente referidos (***'02 - Email Sent*** e ***'03 - WhatsApp Sent'***), através das páginas ***Send E-mail*** *e* ***Send Message*** respetivamente. Na Secção C.5 estão ilustrados exemplos dos *e-mails* e mensagens WhatsApp gerados por este processo. O último passo do processo consiste na inserção do conteúdo do alerta no HIDRA através da página ***Insert in Kibana***.

### 4.3.5 RoboCISO002 - Health Check

Este processo é responsável pela verificação do funcionamento de algumas plataformas e componentes críticas através do Alienvault, do HIDRA e do próprio RoboCISO. O processo implementado está ilustrado nas figuras 4.14 e 4.15.

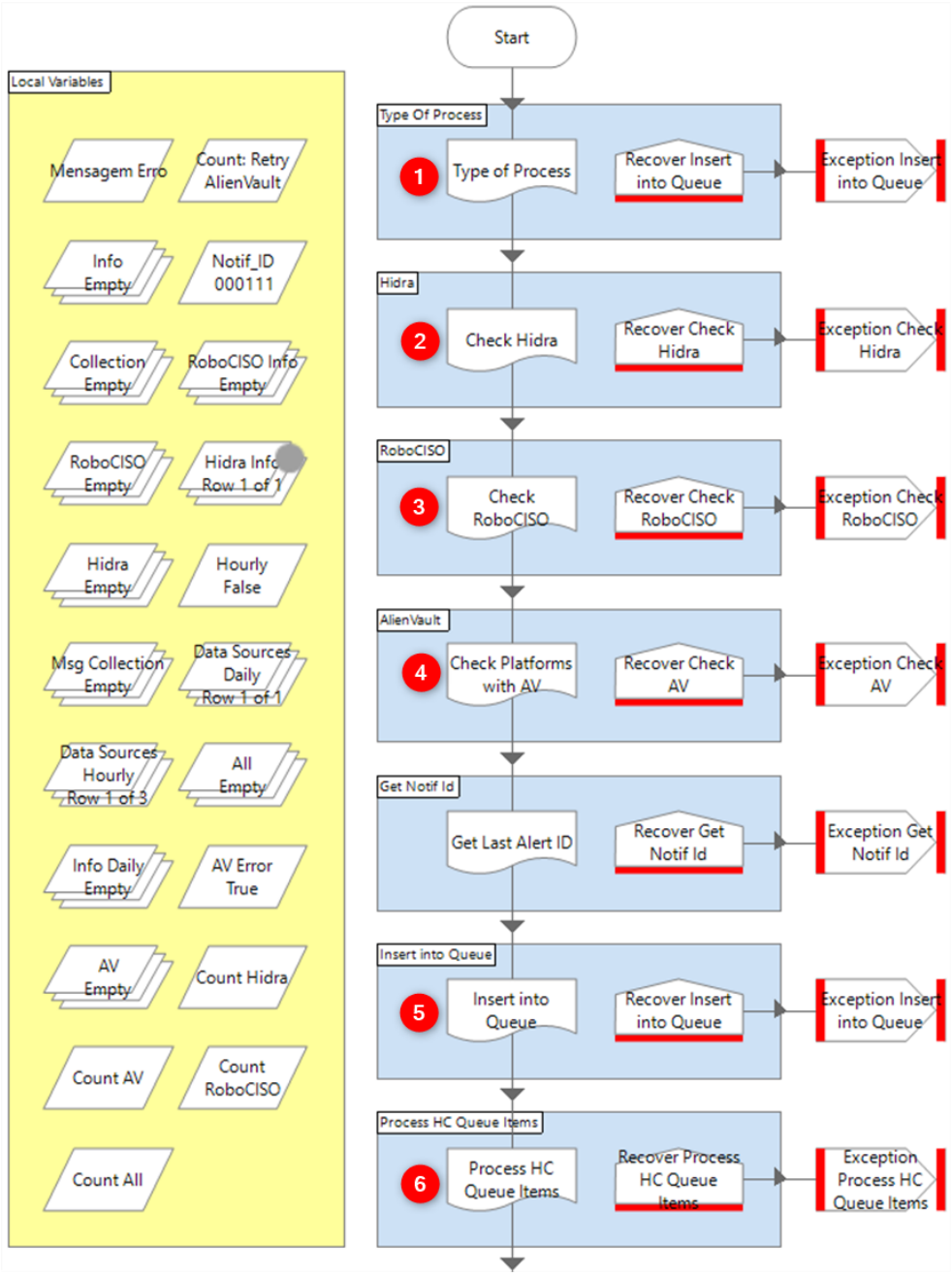



Figura 4.14: Primeira parte da *Main Page* do processo *Health Check*.

Uma vez que este processo tem uma versão horária e outra horária e diária, o primeiro passo ***Type of Process*** (1) verifica qual das versões irá ser executada. Esta decisão tem como base a hora de execução do processo, se a hora estiver entre as 10:15h e 10:45h será do tipo horário e diário retornando a *flag Hourly* a *False*, caso contrário retornará *True*. De seguida é verificado o funcionamento do HIDRA, tal como descrito na Subseção 3.5.5, através da página ***Check HIDRA*** (2) e depois o RoboCISO, ***Check Robo-***

***CISO*** (3). Por fim são verificadas as plataformas/componentes através do Alienvault, ***Check Platforms with AV*** (4). Para cada uma destas plataformas é recolhida a informação do último evento registado e o número total de eventos na última hora ou dia, dependendo da plataforma. Os campos obtidos estão descritos na Tabela 4.3.

| Campo | Descrição |
|---|---|
| *Datasource* | Nome da plataforma/componente |
| *Event Name* | Nome do evento |
| *Timespan* | Data e hora do evento |
| *Volume* | Número total de eventos na última hora ou último dia |

Tabela 4.3: Campos obtidos sobre cada plataforma verificada através do Alienvault.

O *Id* do novo alerta é obtido através da ***Get Last Alert ID***, que será necessário para criar a *Item Key* dos itens, sendo por isso obtido mesmo que não exista um alerta a enviar. Após todas as plataformas/componentes terem sido verificadas, são inseridas na fila de trabalho *RoboCISO002 - Health Check*, através da página ***Insert into Queue*** (5), com uma *Item Key* com a estrutura *Health Check#Id#Platform*, por exemplo *Health Check#78#VPN*, com o *status* ***'01 - Inserted'*** e a *tag* com a estrutura ***Platform - Status***, por exemplo *VPN - Not Working*. Na página ***Process HC Queue Items*** (6), os itens adicionados no passo anterior à fila de trabalho *RoboCISO002 - Health Check* são iterados, sendo verificado se a sua *tag* contém ***Not Working***:

- Se contém, o seu *status* é atualizado para ***´02 - Processed'***, marcado como *Completed* e adicionado à fila de trabalho *RoboCISO002 - Alerts* com a mesma *item key, tag* e *status* ***'01 - Alert Detected'***;

- Se não contém, o seu *status* é atualizado para ***´02 - Processed'*** e marcado como *Completed*.

Na segunda parte da *Main Page*, na página ***Get Alerts*** (7), são obtidos os itens pendentes referentes a alertas *Health Check* da fila de trabalho de alertas. Desta página são devolvidas quatro coleções distintas devido à diferença entre os campos das plataformas/componentes verificadas: *HIDRA, RoboCISO, All* que possui a informação de todos os itens referentes a plataformas verificadas através do Alienvault e, a coleção *AV* que é criada apenas no caso do *Alienvault* esteja inacessível impedindo a verificação das componentes. É feita uma contagem do número de linhas de cada uma dessa coleções, ***Count HIDRA, Count RoboCISO, Count All*** e ***Count AV***, se todas as coleções estiverem vazias, verificado em ***All collections empty?*** (8), não existem alertas e a execução termina. Se pelo menos uma das coleções contém elementos, o próximo passo na execução corresponde ao ***Send Email***, semelhante ao dos restantes processos. O *status* dos itens enviados no *e-mail* é atualizado para ***'02 - Email Sent'***.

Após o envio do *e-mail* é gerada uma coleção reduzida que junta todas as coleções anteriores numa só de forma a compactar a informação a ser enviada pelo WhatsApp, através da página ***Create Msg Collection***. A mensagem é enviada através da página ***Send Message*** e o *status* dos itens é atualizado para ***'03 - WhatsApp Sent'***. Na Secção C.6 estão ilustrados exemplos dos *e-mails* e mensagens WhatsApp gerados por este processo. Por fim o conteúdo da coleção *Msg* é inserido no HIDRA, através da página ***Insert in Kibana***.

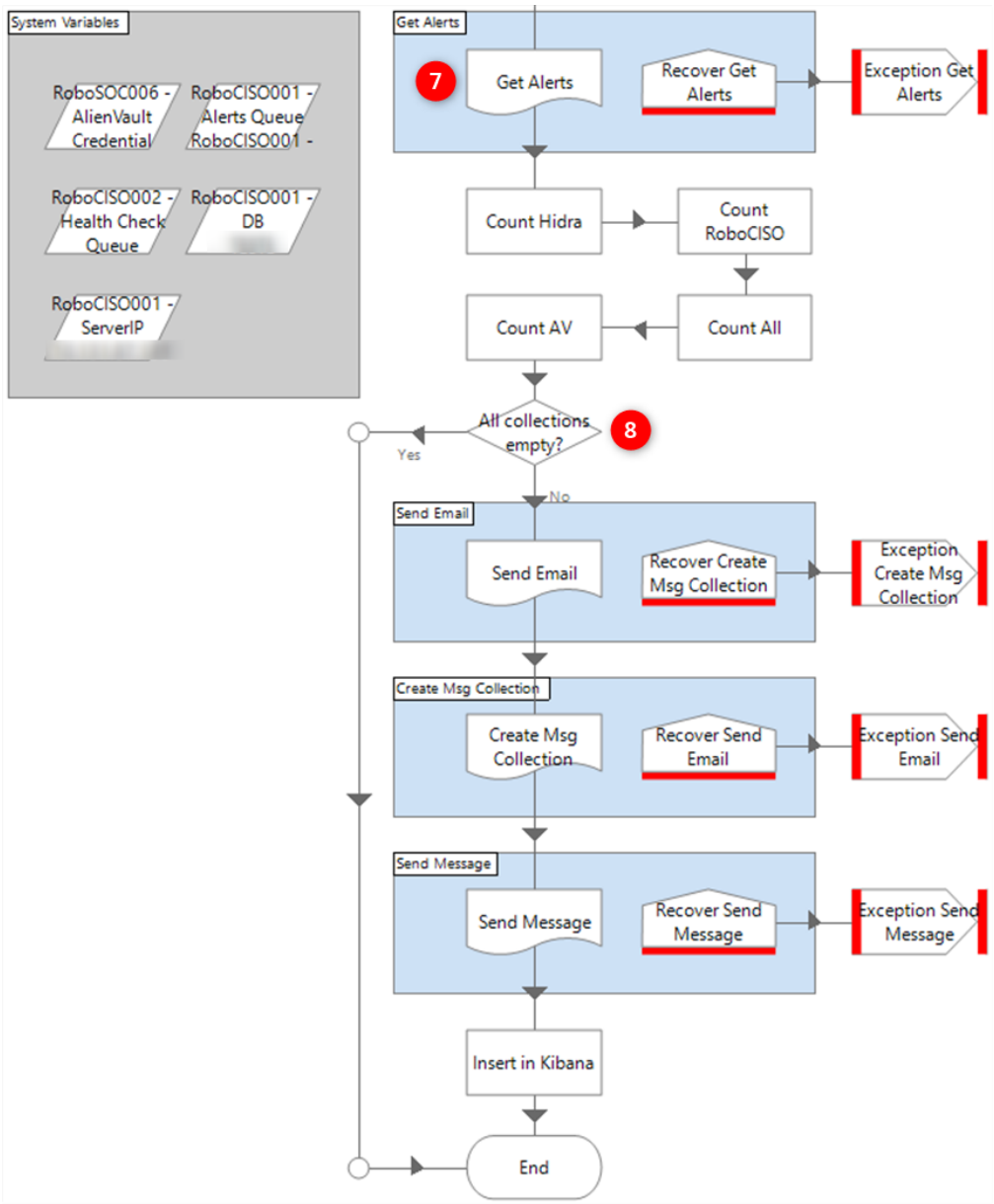



Figura 4.15: Segunda parte da *Main Page* do processo *Health Check*.

# Capítulo 5

# Avaliação e Resultados

Os capítulos anteriores descreveram em detalhe os objetivos, a arquitetura e a implementação do RoboCISO. Este é o capítulo onde é apresentada a avaliação do sistema desenvolvido de forma a perceber se os objetivos foram alcançados. O RoboCISO foi desenvolvido para ser um "assistente virtual" do CISO, informando-o de forma sintetizada e com regularidade acerca dos seus vetores de risco mais críticos e facilitando assim a tomada de decisão. Este capítulo divide-se em duas secções: a Secção 5.1 que avalia a solução de forma a perceber se o sistema apresenta bons indicadores de desempenho que permitam que seja utilizado como sistema principal de envio de notificações e alerta e a Secção 5.2 que avalia a solução em termos qualitativos, ou seja, é feita uma avaliação pelo próprio CISO relativamente à periodicidade e conteúdo das notificações e alertas.

## 5.1 Avaliação Quantitativa

Esta avaliação permite perceber se o RoboCISO tem um bom desempenho de forma a ser utilizado de forma fiável como o assistente virtual do CISO. Foram utilizadas três métricas para avaliar cada um dos processos implementados:

1. **Tempo de execução do processo**: diferença entre o período de conclusão e o período de início do processo;

2. **Percentagem da taxa de erro**: do número total de execuções qual a percentagem que terminou em exceção.

3. **Percentagem de falsos positivos**: do total de alertas/notificações enviados qual a percentagem de falsos positivos.

Os processos foram colocados em produção em datas distintas e todos os resultados serão apresentados com base no tempo que estiveram em produção e o número de execuções nesse período. Sendo necessário fazer testes também em ambiente de produção de forma a garantir que este irá correr sem erros. Desta forma as execuções de teste não serão contabilizadas para o cálculo destas métricas.

### 5.1.1 RoboCISO001 - Critical Patching

As métricas utilizadas para avaliar este módulo são: tempo de execução do processo e percentagem de taxa de erro. Os processos *RoboCISO001 - 01 - Get Microsoft Updates List* e *RoboCISO001 - 02 - Get CVE List from Outlook* foram colocados em ambiente de produção no dia 28 de Maio de 2020 e, os processos *RoboCISO001 - 03 - Get KBs and Severity from each CVE* e *RoboCISO001 - 04 - Classify KBs* no dia 4 de Junho de 2020 tendo o período de observação decorrido até dia 12 de Agosto de 2020 para todos os processos. Este módulo é executado todas as quintas-feiras às 16:00h, começando no processo *RoboCISO001 - 01 - Get Microsoft Updates List* e seguindo a ordem numérica dos processos até ao *RoboCISO001 - 04 - Classify KBs*.

O processo ***RoboCISO001 - 01 - Get Microsoft Updates List*** foi executado 13 vezes sendo que 3 terminaram em exceção, tendo uma taxa de erro de 23%. Este processo tem um tempo de execução médio de 111 segundos, que varia consoante o número de *patches* a ler do sumário da Microsoft. O processo ***RoboCISO001 - 02 - Get CVE List from Outlook*** foi executado 11 vezes tendo 1 terminado em exceção apresentando uma taxa de erro de 9,1% e apresenta uma duração média de 37 segundos.

O processo ***RoboCISO001 - 03 - Get KBs and Severity from each CVE*** foi executado 11 vezes tendo uma 1 delas terminado em exceção perfazendo uma taxa de erro de 9,1%. Este processo possui um tempo de execução médio de 367 segundos. O processo ***RoboCISO001 - 04 - Classify KBs*** foi executado 9 vezes e nenhuma das suas execuções terminou em exceção possuindo uma taxa de erro de 0%. O tempo de execução médio do processo é de 207 segundos, variando com o número de *patches* a classificar.

Nenhum dos processos deste módulo produz alertas/notificações, contudo esta pequena avaliação foi realizada na mesma pois se estes processos não funcionarem corretamente o processo *RoboCISO002 - Patching Alert* não possuirá a informação necessária para cumprir a sua função. Através do tempo médio calculado para cada processo é possível concluir que toda a informação necessária relativamente aos *patches* e vulnerabilidades é obtida de forma rápida, consumindo menos de 15 minutos por semana. O processo *RoboCISO001 - 01 - Get Microsoft Updates List* é aquele que apresenta a maior taxa de erro. A maior parte das execuções que terminaram em exceção aconteceram após a inserção do processo em ambiente de produção. O código continha alguns erros que foram corrigidos e portanto passaram a ser executadas com sucesso. Uma vez que o período de observação foi curto (aproximadamente 10 semanas), será de esperar que com um maior tempo de observação esta taxa de erro diminua, tanto para este processo como para os restantes.

### 5.1.2 RoboCISO002 - Alerts

As métricas utilizadas para avaliar este módulo são: percentagem de taxa de erro e percentagem de falsos positivos. Os cinco processos que constituem o módulo *RoboCISO002 - Alerts* estão divididos em dois horários de execução:

- ***RoboCISO002 - Hourly Alert*** é executado todos os dias, uma vez por hora, das 00:30h às 23:30h, sendo composto pelos processos *RoboCISO002 - DDoS Alert*, *RoboCISO002 - Bitsight Alert* e *RoboCISO002 - Health Check*;
- ***RoboCISO002 - Daily Alerts*** é executado uma vez por dia às 9:10h, sendo composto pelos processos *RoboCISO002 - SLA Exceeded* e *RoboCISO002 - Patching Alert*.

#### RoboCISO002 - Patching Alert

O processo *RoboCISO002 - Patching Alert* foi colocado em ambiente de produção no dia 1 de Maio de 2020 tendo o intervalo de observação decorrido até ao dia 12 de Agosto de 2020. Durante este intervalo de tempo, o número total de execuções foi de 126 sendo que 25 terminaram em exceção. Este processo apresenta uma taxa de erro de $21,9\%$. O principal fator que contribuiu para esta percentagem ser tão elevada foi um erro de conexão com o WhatsApp que causava a falha do processo na altura de enviar a mensagem. O número total de notificações contabilizadas para este processo é de 91, estando todas corretas, não existindo falsos positivos.

#### RoboCISO002 - DDoS Alert

O processo *RoboCISO002 - DDoS Alert* foi colocado em ambiente de produção no dia 8 de Maio de 2020 e o intervalo de observação durou até ao dia 12 de Agosto de 2020. O número total de execuções foi de 2096 sendo que 26 terminaram em exceção. A taxa de erro deste processo é portanto de $1,2\%$, tratando-se de um valor bastante baixo. De todas as execuções apenas 106 foram efetivamente alertas. Devido a um *bug* no código, 12 dos 106 alertas gerados foram falsos positivos tendo sido gerados todos no mesmo dia.

#### RoboCISO002 - SLA Exceeded

O processo *RoboCISO002 - SLA Exceeded* foi colocado em ambiente de produção a partir de dia 9 de Maio de 2020 e o intervalo de observação durou até ao dia 12 de Agosto de 2020. O número total de execuções deste processo foi de 102 sendo que 8 terminaram em exceção, perfazendo uma taxa de erro de $7,3\%$. Foram geradas 41 notificações e devido a um *bug* no código 1 das notificações geradas foi repetida sendo por isso um falso positivo. O problema foi corrigido no próprio dia não voltando a acontecer.

#### RoboCISO002 - Bitsight Alert

O processo *RoboCISO002 - Bitsight Alert* foi colocado em ambiente de produção no dia 29 de Maio de 2020 e o intervalo de observação decorreu até dia 12 de Agosto de 2020. Durante este intervalo de tempo, o número total de execuções foi de 1636 sendo que 38 terminaram em exceção, apresentando uma taxa de erro de $2,3\%$. Apenas existiu um alerta gerado por este processo e tratou-se de um verdadeiro positivo.

#### RoboCISO002 - Health Check

O processo *RoboCISO002 - Health Check* foi colocado em ambiente de produção no dia 5 de Junho de 2020 e o intervalo de observação durou até dia 12 de Agosto de 2020. O número total de execuções foi de 1279 sendo que 260 terminaram em exceção, apresentando uma taxa de erro de $20,3\%$. As exceções aconteceram em apenas algumas das execuções não sendo por isso fácil descobrir a sua origem, o que explica a taxa de erro tão elevada. A maioria dos erros teve origem em três problemas que foram entretanto solucionados:

1. A página do Alienvault (o SIEM) por vezes é considerada como insegura não deixando o robô aceder à sua interface. Após identificar este problema foram adicionados passos extra para lidar com a situação;

2. Numa das plataformas verificadas (*Cyberark*) o resultado das procuras não carregava chegando a exceder 6 minutos de espera causando a falha do processo. Após este problema ser identificado a única solução encontrada foi remover o *Cyberark* das plataformas a verificar;

3. Para ser possível aceder à interface do Alienvault foi necessário criar uma conta para o RoboCISO. Acontece que a conta tinha algumas permissões em falta e devido ao número de vezes que acedia à interface e ao número de *queries* executadas, este ficava sobrecarregado o que causava a indisponibilidade da interface, resultando numa falha do processo. Para além da indisponibilidade da interface, por vezes as procuras não apresentavam resultados, apresentando apenas avisos, como por exemplo "*Unable to query datasource*", com os quais o robô não estava preparado para lidar causando também uma exceção. Foram identificadas as possíveis mensagens e adicionados passos para lidar com elas.Após perceber que a ausência de algumas permissões da conta era a causa destes problemas, estas foram atribuídas e estes problemas foram ultrapassados.

Numa primeira versão deste processo eram gerados alertas se o número de eventos estivesse abaixo da média para essa hora e dia da semana. Esta estratégia gerou um excesso de alertas o que levou à alteração da condição inicial, sendo agora gerado um alerta apenas se o número de eventos for igual a zero. Desta forma dos 122 alertas, apenas 29 corresponderam a alertas que reportaram 0 eventos, para pelo menos uma das plataformas verificadas. Esses 29 alertas foram gerados, em grande parte, devido aos problemas descritos no Item 3 da listagem de erros, pelo que correspondem a falsos positivos, apresentado assim uma taxa de 100% de falsos positivos. Outro fator que contribuiu para estes falsos positivos, foi causado devido a má decisão de implementação, onde se assumia que nas situações em que a interface do Alienvault estava inacessível, as plataformas a verificar apresentavam 0 eventos, o que poderia não corresponder à realidade.

## 5.2 Avaliação Qualitativa

Sendo o CISO o utilizador primário desta solução, a avaliação em termos qualitativos é feita por este. Após a implementação de todos os casos de uso, o RoboCISO entrou em período de testes. Decorrido este período foi feita a verificação do cumprimento ou não dos requisitos estipulados:

1. As notificações e alertas devem ter uma periodicidade adequada de forma a evitar a *alert fatigue*;

2. As notificações e alertas devem conter toda a informação relevante incluída e devidamente assinalada assegurando que notificações/alertas relativos ao mesmo tema são enviados numa única mensagem/*e-mail* sempre que possível;

3. Os alertas referentes a: ataques DoS/DDoS; alterações de *ratings Bitsight* ou *Health Checks* devem ser reportados em tempo útil;

Após o *feedback* do CISO foi verificado se o implementado atingiu os objetivos propostos:

- O **ponto 1** foi parcialmente cumprido uma vez que para o processo Health Check o CISO considera ter gerado um excesso de informação. Este excesso de informação deveu-se ao facto da primeira condição para alerta ser baseada na média de eventos. Para os casos dos processos *Bitsight Alert*, *SLA Exceeded* e *Patching Alert* o CISO considera que têm uma periodicidade adequada. No caso do processo *DDoS Alert* a sua periodicidade poderá necessitar de alterações.

- O **ponto 2** também foi parcialmente cumprido uma vez que para cada tipo distinto de alerta é enviada uma única mensagem, independentemente do número de itens a reportar, por exemplo se existirem três *tickets* cujo SLA foi expirado serão os três reportados numa única mensagem WhatsApp. No caso do *DDoS Alert* o CISO quer mais informação sobre o alvo do ataque. Atualmente é apenas enviado o IP do alvo tal como apresentado pela própria plataforma de mitigação, o que o CISO considerou insuficiente;

- O **ponto 3** foi também cumprido uma vez que os processos de *DDoS Alert, Bitsight Alert* e *Health Check* são executados uma vez por hora. Este agendamento permite que quaisquer eventos, relacionados com estes processos, cheguem ao conhecimento do CISO em menos de uma hora.

## 5.3 Discussão dos resultados

Na tabela Tabela 5.1 está um resumo com os valores obtidos para cada processo em cada métrica: o número total de execuções; número total de erros, ou seja, execuções que terminaram em exceção; a taxa de erro; número de falsos positivos; percentagem de falsos positivos e o número de alertas/notificações geradas.

| Processo | # Execuções | # Exceções | % Erro | # Alertas/Notificações | # FP | % FP |
|---|---|---|---|---|---|---|
| *Patching A.* | 114 | 25 | 21,9 | 91 | 0 | 0 |
| *DDoS A.* | 2096 | 26 | 1,2 | 106 | 12 | 11,3 |
| *SLA Exc.* | 102 | 8 | 7,3 | 41 | 1 | 2,4 |
| *Bitsight A.* | 1636 | 38 | 2,3 | 1 | 0 | 0 |
| *Health Check* | 1279 | 260 | 20,3 | 122 | 122 | 100 |

Tabela 5.1: Tabela sumária do desempenho dos processos do módulo *RoboCISO002 - Alerts*.

Analisando estes resultados podemos concluir que os processos *RoboCISO002 - DDoS Alert* e *RoboCISO002 - Bitsight Alert* apresentam taxas de erro muito baixas, estando por isso em condições para serem considerados fiáveis como fontes de alertas. Apesar de no caso do *RoboCISO002 - DDoS Alert* existir uma taxa de falsos positivos de 11,3%, estes aconteceram pouco depois da inserção do processo em ambiente de produção devido a um erro no código que foi de imediato resolvido, sendo por isso de esperar que esta percentagem diminua.

Já o processo *RoboCISO002 - Patching Alert* apresenta a maior taxa de erro. O principal erro estava relacionado com o estabelecimento da conexão entre o Blue Prism e o Chrome quando era necessário enviar a notificação diária via WhatsApp. A causa do erro estava relacionada com o número de processos que podem ser executados em simultâneo no Blue Prism. Se a execução do processo demorar mais que

o normal, pode ocorrer a interseção de cinco processos a serem executados em simultâneo, quando atualmente o número de licenças do Blue Prism permite apenas a execução simultânea de quatro processos. A calendarização dos processos foi entretanto ajustada e será de esperar que a taxa de erro diminua.

O processo *RoboCISO002 - SLA Exceeded* enviou apenas um falso positivo e tem uma taxa de erro de 7,3%, sendo ainda um pouco elevada. Este processo é executado após o *RoboCISO002 - Patching Alert*, que apresenta uma taxa de erro elevada e por vezes ao falhar não deixou o ambiente em condições do processo de SLA ser executado causando a sua falha, por exemplo: se uma página do Chrome ficar aberta após a falha do processo de *patching* e se o processo de SLA precisar de enviar uma notificação, este irá abrir uma nova página e o robô não saberá qual delas utilizar, o que causará a falha do processo. Uma vez que muitos dos erros deste processo não foram causados pelo próprio, mas sim devido a conflitos causados pela falha do *RoboCISO002 - Patching Alert* será de esperar que com um período de observação mais longo, a taxa de erro diminua.

Por fim, o processo *RoboCISO002 - Health Check* apresenta também uma taxa de erro elevada, cerca de 20,3%. Esta taxa deveu-se principalmente aos erros descritos na Secção 5.1.2. Será de esperar que após um período de observação superior esta taxa também diminua uma vez que os erros mais frequentes e conhecidos foram resolvidos. Devido à condição de alerta definida inicialmente todos os alertas gerados corresponderam a falsos positivos. A alteração da condição para alerta ocorreu a 15 de Julho de 2020 pelo que, o período de observação ainda é muito curto. Se for realizada uma nova avaliação dentro de alguns meses poderá mostrar melhores resultados.

Relativamente à avaliação qualitativa feita por parte do CISO, de modo geral, o conteúdo e periodicidade dos alertas e notificações encontram-se de acordo com o esperado. Contudo, precisam de algumas afinações como por exemplo, o conteúdo no processo *DDoS Alert* e o processo de *Health Check* precisa de ficar bem parametrizado de modo a evitar o excesso de alertas.

Se existir um novo ciclo de desenvolvimento neste projeto:

- O processo *RoboCISO002 - Health Check* apresentou maus resultados e por isso tem de ser acompanhado de perto para perceber se as alterações feitas melhoraram ou não o seu desempenho. Muitos problemas foram mitigados e a condição de alerta foi alterada contudo, o período de observação ainda é muito curto (menos de um mês). Fazendo uma nova avaliação dentro de alguns meses e tendo em conta apenas os alertas gerados desde a nova condição ter sido aplicada poderá mostrar se o processo é viável ou se terá de ser redesenhado/reimplementado;

- Melhorar o conteúdo das mensagens como por exemplo no processo *RoboCISO002 - DDoS Alert* onde o CISO considerou que a informação sobre o alvo do ataque é reduzida/insuficiente;

- Para processos que geram notificações de forma a evitar o envio de várias mensagens seguidas, estas poderão ser enviadas em conjunto numa única mensagem. Atualmente só existem dois processos que geram notificações (*Patching* e *SLA Exceeded*) mas, se forem adicionados novos vetores críticos que produzam também notificações isso causará um excesso de mensagens e *e-mails*. Desta forma os processos de notificações atualmente existentes poderão ficar responsáveis apenas pela obtenção e deteção das situações a reportar e um novo processo responsável pelo envio da notificação com todo o conteúdo;

- Incluir uma componente de visualização nos relatórios. Um aspeto muito importante e não abordado nesta primeira iteração do sistema é a visualização de dados. Atualmente a informação é enviada organizada em tabelas, quando enviada por *e-mail* ou texto quando enviado por Whatsapp. Esta não é a melhor forma de apresentar os dados, principalmente se forem adicionados mais vetores e se se pretender um relatório mais elaborado. Num novo ciclo de desenvolvimento será imperativa a adição da componente de visualização de dados permitindo uma análise mais rápida da informação bem como, a análise de uma maior quantidade de informação recorrendo a um espaço mais pequeno (*e.g.* imagem/gráfico). Esta primeira iteração serviu principalmente para perceber o potencial ou não da solução bem como dos diferentes casos de uso implementados.

# Capítulo 6

# Conclusão

Neste trabalho, foi apresentado o RoboCISO, uma solução baseada em RPA que visa manter o CISO informado em relação ao estado dos seus vetores mais críticos. Tratando-se da primeira versão, este projeto focou-se num pequeno conjunto de cinco vetores críticos (*Patching*, SLA, ataques de DoS/DDoS, *Health Checks* e *Bitsight ratings*). O RoboCISO mantém o CISO informado em tempo útil relativamente ao estado destes vetores gerando notificações regulares (no caso do *Patching* e do SLA) e alertas sempre que necessário (no caso de ataques de DoS/DDoS, *Health Checks* às plataformas críticas e alterações acentuadas nos *Bitsight ratings*) através do *e-mail* e do WhatsApp.

O sistema desenvolvido apresenta algumas limitações: a primeira deve-se ao próprio *software* onde foi desenvolvido, o Blue Prism. O facto de requerer a interação com várias plataformas *web*, é necessária uma interação com elementos específicos dessas plataformas. Se existirem grandes alterações na interface de utilizador os processos poderão deixar de funcionar sendo necessário identificar de novo esses elementos e possivelmente alterar a lógica do processo. Outra limitação é a interação com o WhatsApp, que não sendo uma versão comercial necessita de um telemóvel sempre ligado e conectado à Internet para que seja possível enviar as mensagens.

Relativamente à avaliação da solução, apresentada no Capítulo 5, é possível concluir que alguns dos processos precisam de mais tempo e afinação para serem considerados fiáveis como fonte primária dos alertas e notificações, sendo estes o *Patching Alert* e o *Health Check*. Estes processos possuem taxas de erro muito elevadas tendo por isso de ser acompanhados com regularidade para perceber a origem dos problemas e solucioná-los. Os processos de *DDoS Alert* e *Bitsight Alert* apresentam os melhores desempenhos, com taxas de erro e taxas de falsos positivos muito baixas, sendo por isso aqueles que já apresentam condições para serem a fonte primária para os alertas destes vetores. O processo de SLA se mantiver o bom desempenho atual irá entrar também no grupo de processos considerados fiáveis como fonte primária de alertas.

Apesar das limitações referidas e de alguns processos ainda apresentarem problemas de conteúdo e/ou taxas de erro e/ou taxas de falsos positivos um pouco elevadas, os resultados apresentados mostram que pode e deve ser considerado como uma solução exequível e fiável para manter o CISO informado sobre o estado dos seus vetores mais críticos. O trabalho apresentado neste relatório deu origem à versão inicial do RoboCISO, do qual se perspetivam evoluções. Desta forma, os resultados são promissores, sendo que no futuro adquirindo uma versão comercial através de uma API do WhatsApp pode tornar o RoboCISO ainda mais robusto e numa solução de melhor qualidade.

## 6.1 Trabalho Futuro

A solução implementada corresponde à primeira versão do RoboCISO sendo por isso desenvolvida tendo como foco um pequeno conjunto de vetores críticos. Futuras versões deste projeto deverão ser mais orientadas à *situational awareness* abordada no Capítulo 2. Esta versão do RoboCISO serviu assim para compreender o potencial da solução. Alguns dos pontos a abordar no futuro poderão ser:

- Criação de um relatório de síntese/*briefing* para que o CISO possa começar o seu dia com toda a informação situacional que lhe seja relevante. Neste relatório poderá estar incluído um resumo dos vetores críticos analisados, por exemplo: no caso do DDoS, utilizado atualmente apenas na vertente de alertas, poderá ser também inserido neste relatório num resumo dos últimos 10 ataques de DoS/DDoS mais relevantes. As notificações de *Patching* e *SLA* poderão fazer parte deste relatório matinal, estando contidos numa única mensagem de forma a diminuir o número de vezes que o CISO recebe mensagens;

- Adicionar novos tipos de alerta, por exemplo para ciber ataques relevantes em curso. Neste momento são apenas reportados os ataques de Dos/DDoS mas poderão ser adicionados outros tipos de ataque como *brute force*. Adicionar novas plataformas ao processo de *Health Check* trará também uma visão cada vez mais global do estado real dos sistemas da infraestrutura, dado que atualmente são verificadas apenas cinco plataformas críticas descritas na Subseção 4.3.5;

- Possibilitar a resposta automática do RoboCISO a determinados comandos como por exemplo: enviando o código *#ddos_week* via WhatsApp este poderia enviar automaticamente um relatório dos últimos sete dias com um resumo dos ataques DDoS decorridos nesse período.

# Abreviaturas

**BP** Blue Prism.

**CISO** Chief Information Security Officer.

**CSOC** Cyber Security Operations Center.

**CVE** Common Vulnerabilities and Exposures.

**DDoS** Distributed Denial of Service.

**DoS** Denial of Service.

**KB** Knowledge Base.

**MDR** Managed Detection and Responses.

**MSSP** Managed Security Service Providers.

**RPA** Robotic Process Automation.

**RTIR** Request Tracker Incident Response.

**SA** Situational Awareness.

**SIEM** Security Information and Event Management.

**SLA** Service Level Agreement.

**SOC** Security Operations Center.

# Bibliografia


[1] DCY: *"Política de Segurança da Informação da Altice Portugal"*. página 6, julho 2019.

[2] LLC, Ponemon Institute: *"The impact of data breaches on reputation & share value"*. página 12, maio 2017.

[3] Fruhlinger, Josh: *"What is a CISO? Responsibilities and requirements for this vital leadership role"*. https://www.csoonline.com/article/3332026/what-is-a-ciso-responsibilities-and-requirements-for-this-vital-leadership-role.html, janeiro 2019.

[4] Alegria, José: *"Proteção Adicional Contra Ataques Agressivos de Malware"*. INGENIUM, (159):59–61, junho 2017.

[5] Purplesec: *"2020 Ransomware Statistics, Data, & Trends"*. https://purplesec.us/resources/cyber-security-statistics/ransomware/, 2020.

[6] Pina, M.G: *"Automatic Detection of Anomalous User Access Patterns to Sensitive Data"*. página 9, 2019.

[7] Stephanie Overby, Lynn Greiner e Lauren Gibbons Paul: *"What is an SLA? Best practices for service-level agreements"*. https://www.cio.com/article/2438284/outsourcing-sla-definitions-and-solutions.html, julho 2017.

[8] CriticalStart: *"The Impact of Security Alert Overload"*. páginas 3–7, 2019.

[9] Carfagno, Don: *"What Is a Security Patch?"*. https://www.blackstratus.com/what-is-a-security-patch/, agosto 2019.

[10] Souppaya, Murugiah e Karen Scarfone: *"Guide to Enterprise Patch Management Technologies"*. julho 2013.

[11] FIRST: *"Common Vulnerability Scoring System SIG"*. https://www.first.org/cvss/, 2017.

[12] Wilcox, John: *"Windows 10 update servicing cadence"*. https://techcommunity.microsoft.com/t5/windows-it-pro-blog/windows-10-update-servicing-cadence/ba-p/222376, janeiro 2018.

[13] Hoffman, Chris: *"What Is Patch Tuesday for Windows, and When Is It?"*. https://www.howtogeek.com/443161/what-is-patch-tuesday-for-windows-and-when-is-it/, outubro 2019.

[14] Alexandre Lourinho, Alfredo Soares e Carlos Maximiano: *" Política De Distribuição De Updates de Segurança (Estações de Trabalho Windows e Linux)"*. páginas 4–6, julho 2017.

[15] Microsoft: *"Security Update Severity Rating System"*. https://www.microsoft.com/en-us/msrc/security-update-severity-rating-system?SilentAuth=1&wa=wsignin1.0.

[16] MITRE: *"Situation Awareness"*. https://www.mitre.org/capabilities/cybersecurity/situation-awareness.

[17] National Security Systems, Committee on: *"Committee on National Security Systems (CNSS) Glossary"*. abril 2015.

[18] Horneman, Angela: *"Situational Awareness for Cybersecurity: An Introduction"*. https://insights.sei.cmu.edu/sei_blog/2019/09/situational-awareness-for-cybersecurity-an-introduction.html, setembro 2019.

[19] Endsley, M.R.: *"Toward a Theory of Situation Awareness in Dynamic Systems"*. *Human Factors Journal*, year = 1995, pages = 8–10,.

[20] Baldoni, John: *"Situational Awareness 101"*. https://www.cio.com/article/2437930/situational-awareness-101.html, outubro 2007.

[21] Salomaa, Jyri: *"Measuring and Creating Situational Awareness in Cybersecurity: The Requirements Specification for Situational Awareness and Metrics Platform"*. páginas 8–10, dezembro 2019.

[22] Chappel, David: *Introducing blue prism*, maio 2017.

[23] WhatsApp: *End-to-end encryption - security by default*. https://www.whatsapp.com.

[24] Ong, Thuy: *"WhatsApp launches a separate app for small businesses"*. https://www.theverge.com/2018/1/19/16908810/whatsapp-business-app-launch-small-businesses, janeiro 2018.

[25] MITRE: *"What is CVE?"*. https://cve.mitre.org/about/faqs.html.

[26] CISA: *"Security Tip (ST04-015) - Understanding Denial-of-Service Attacks"*. https://www.us-cert.gov/ncas/tips/ST04-015, novembro 2009.

[27] BitSight: *"Four Data Categories in BitSight's Security Ratings Platform"*. https://www.bitsight.com/data.

[28] Kobialka, Daniel: *"What is a SIEM and what are the benefits for business?"*. https://cybersecurity.att.com/blogs/security-essentials/siem-what-is-it-and-why-does-your-business-need-it, junho 2020.

[29] *Dr. Dobb's, The World of Software Development*: *"SIEM: A Market Snapshot"*. https://www.drdobbs.com/siem-a-market-snapshot/197002909, fevereiro 2007.

[30] Pratt, Mary K.: *"What is SIEM software? How it works and how to choose the right tool"*. https://www.csoonline.com/article/2124604/what-is-siem-software-how-it-works-and-how-to-choose-the-right-tool.html, novembro 2017.

[31] Rouse, Margaret: *"DHCP (Dynamic Host Configuration Protocol)"*. https://searchnetworking.techtarget.com/definition/DHCP, dez 2019.

[32] IEEE: *"802.1X-REV - Revision of 802.1X-2004 - Port Based Network Access Control"*. http://www.ieee802.org/1/pages/802.1x-rev.html.

[33] Logbinder: *"Supercharger for Windows Event Collection - Supercharger Free"*. https://www.logbinder.com/Products/Supercharger/.

# Anexo A

# Queries SQL

## A.1 Stored Procedure Classify KBs

```
UPDATE [KB_list] SET Severity=NULL

UPDATE kb SET Severity='Critical'
        FROM [KB_list] kb
        JOIN [CVE_KB] ck ON(ck.Article=kb.Article
            AND ck.Severity='critical'
            AND kb.severity IS NULL)
UPDATE kb SET Severity='Important'
    FROM [KB_list] kb
        JOIN [CVE_KB] ck ON(ck.Article=kb.Article
            AND ck.Severity='important'
            AND kb.severity IS NULL)
UPDATE kb SET Severity='Moderate'
        FROM [KB_list] kb
        JOIN [CVE_KB] ck ON(ck.Article=kb.Article
            AND ck.Severity='Moderate'
            AND kb.severity IS NULL)
UPDATE kb SET Severity='Low'
        FROM [KB_list] kb
        JOIN [CVE_KB] ck ON(ck.Article=kb.Article
            AND ck.Severity='Low'
            AND kb.severity IS NULL)
```

## A.2 Query de obtenção dos alertas Patching Alert

```
SELECT * FROM critical_patching
WHERE ((days_delay>28 and ``%required">10)
OR (days_delay>21 AND ``%required">25)
OR (days_delay>14 AND ``%required">50))
AND #total >(CASE WHEN chassis_type='server'
OR chassis_portable IS NULL
OR chassis_portable='true' THEN 50 ELSE 100 END)
ORDER BY [days_delay] DESC, #required DESC
```

## A.3 Query de obtenção dos *tickets* de DDoS

```
SELECT source=CASE WHEN subject LIKE
    'Arbor Alert #%', ``CF_{Priority}", count(*)
FROM RTIR_incidents
WHERE subject LIKE 'Arbor Alert #%'
AND Created_Dt>'2019-12-01'
GROUP BY CASE WHEN subject LIKE
    'Arbor Alert #%', ``CF_{Priority}"
ORDER BY 1, 2 DESC
```

## A.4 Query de obtenção dos *tickets* com SLA excedido

```
SELECT * from Extract_RTIR_SOC_incidents
WHERE "CF_{Constituency}"='PT'
AND DATEDIFF(hour, Created_Dt,
    (CASE WHEN Resolved_Dt='1970-01-01'
    THEN getdate() ELSE Resolved_Dt END)) > LEFT(SLA,LEN(SLA)-1)
```

# Anexo B

# Código Blue Prism

## B.1 Geração do corpo do *e-mail*

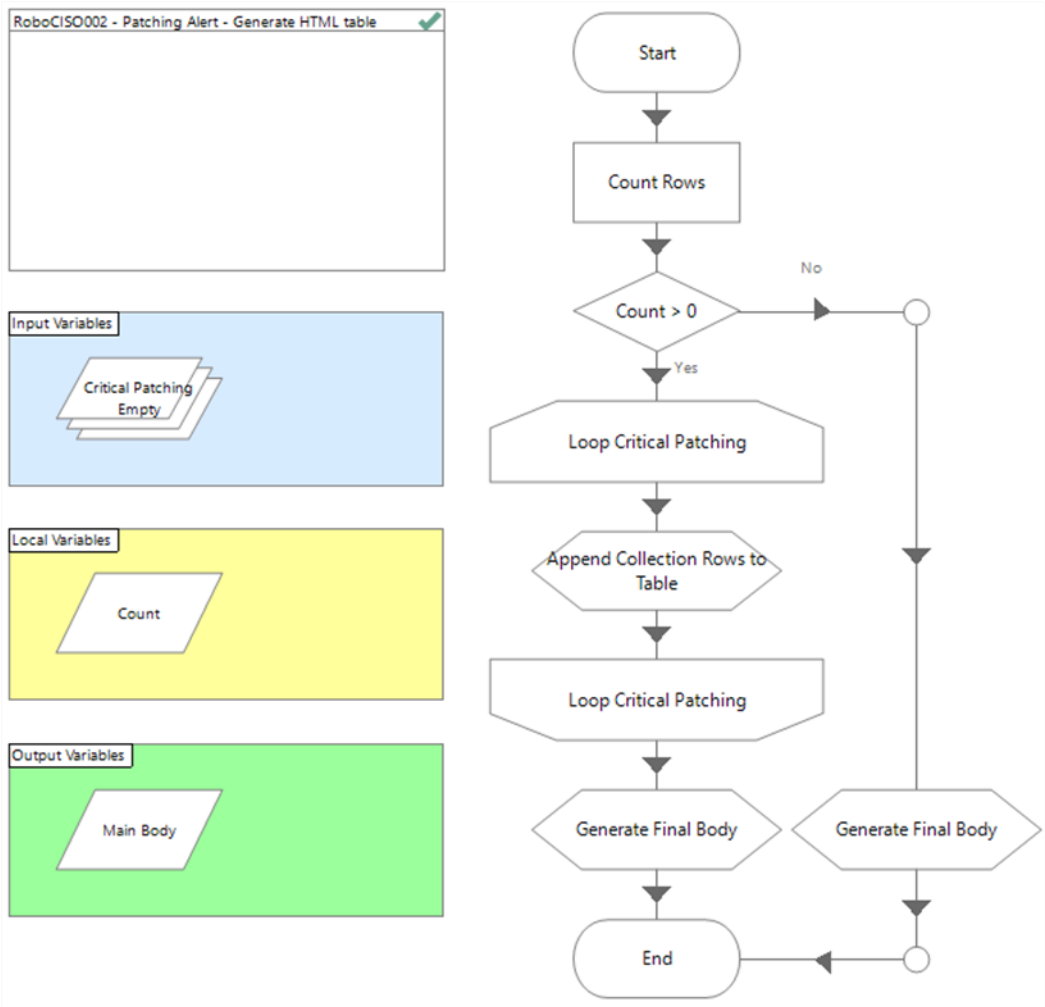



Figura B.1: Etapa de criação da mensagem a ser enviada por *e-mail*.

## B.2 Patching Alert - Send Email

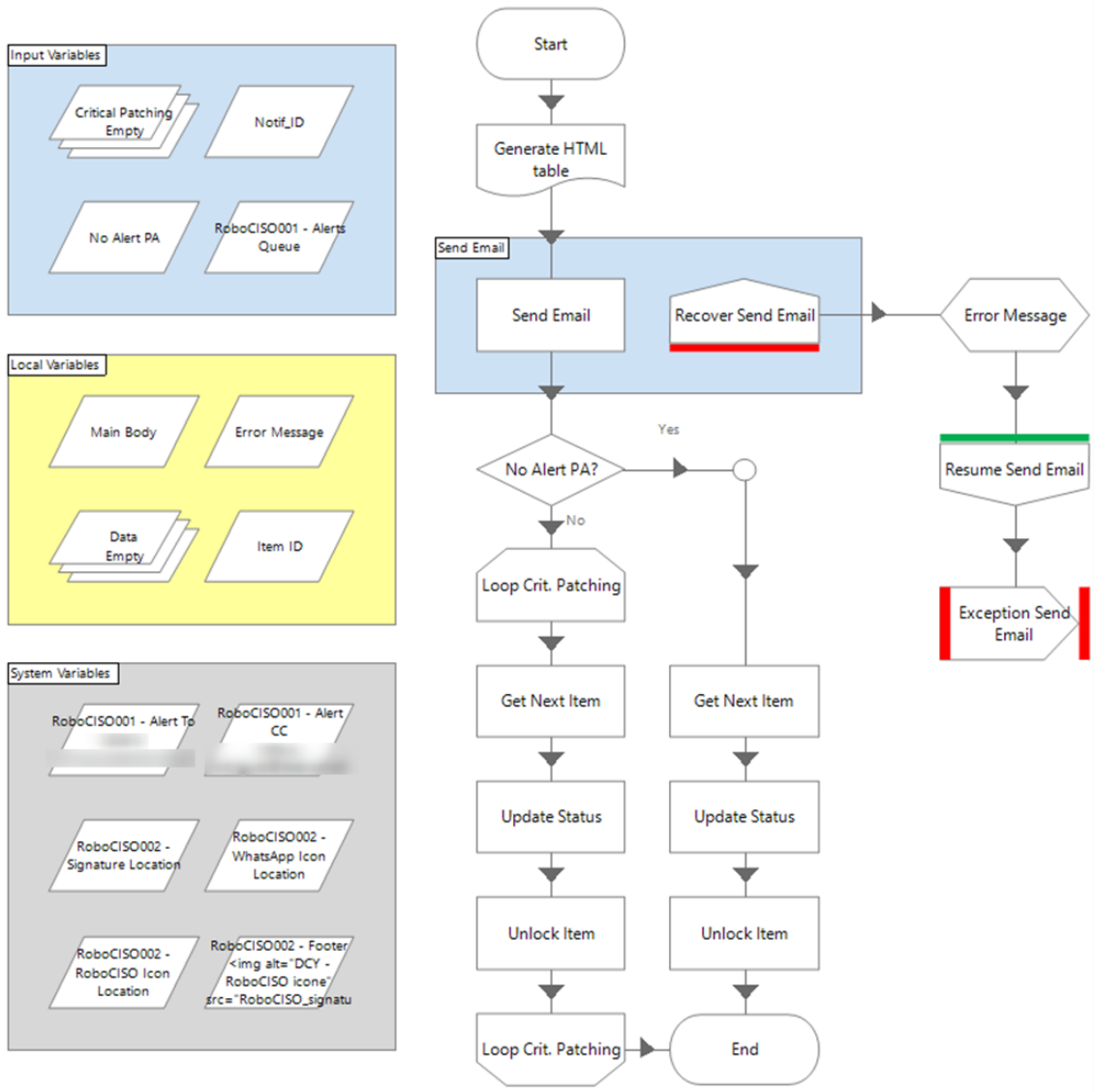



Figura B.2: Etapa de envio do *e-mail*.

## B.3 Patching Alert - Send Message

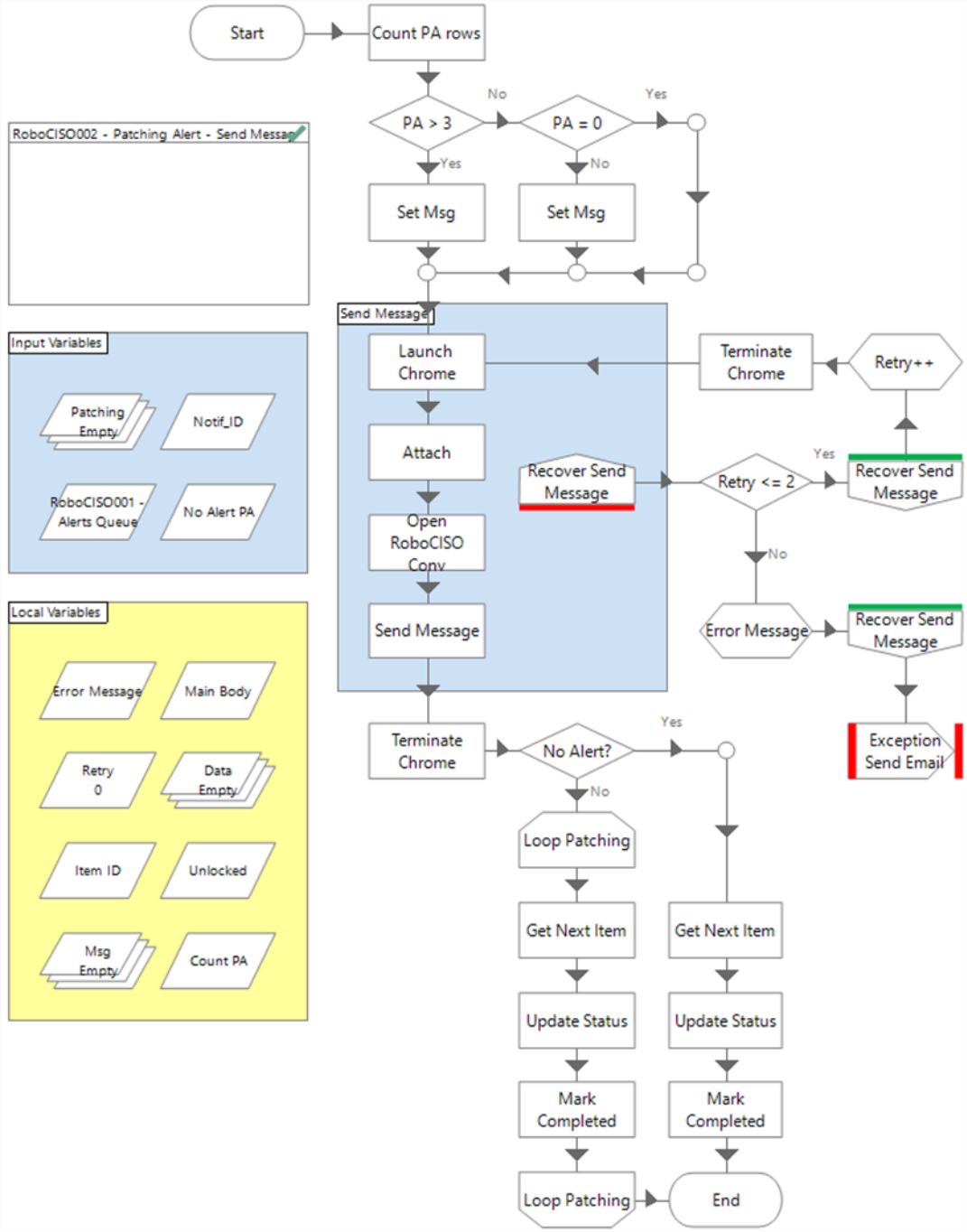



Figura B.3: Etapa de envio da mensagem via WhatsApp.

# Anexo C

# *E-mails* e mensagens WhatsApp

## C.1 Catálogo WhatsApp

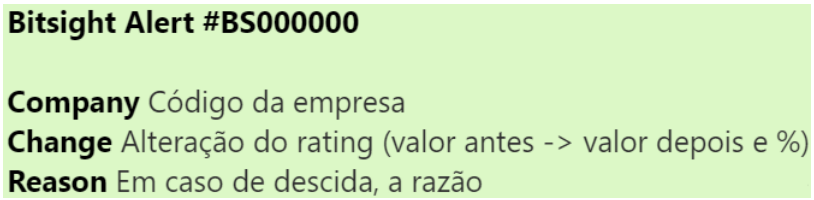



Figura C.1: Exemplo de entrada no catálogo do WhatsApp.

## C.2 Patching Alert

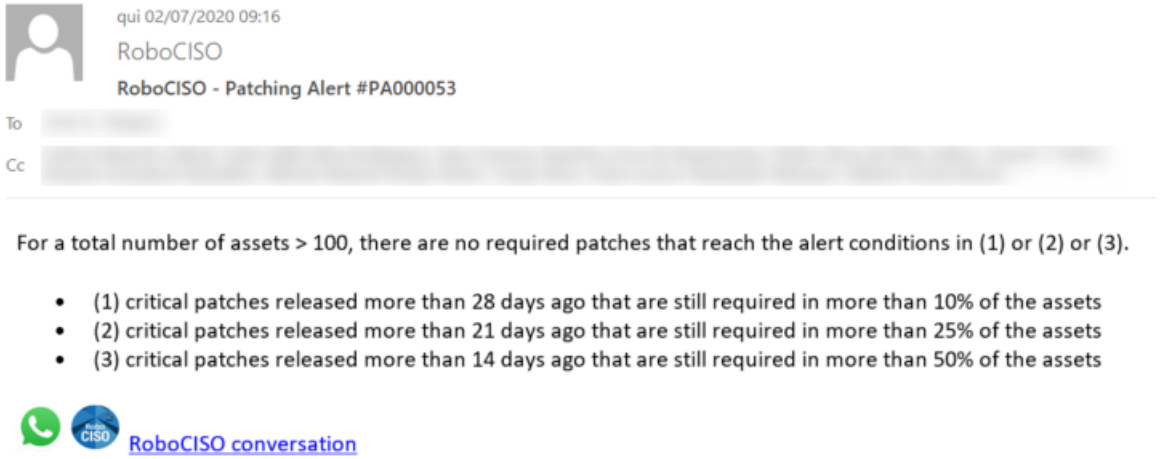



Figura C.2: Exemplo de *e-mail* do *Patching Alert* caso não existam atrasos a reportar.

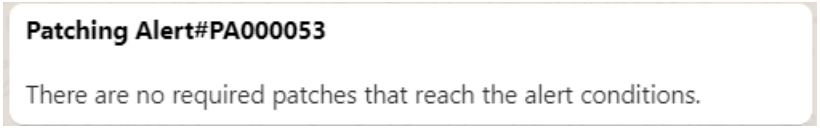



Figura C.3: Exemplo de mensagem do *Patching Alert* caso não existam atrasos a reportar.

qua 24/06/2020 09:41

RoboCISO

**RoboCISO - Patching Alert #PA000045**

To [redacted]

Cc [redacted]

| KB Date | Days Delay | Patch | Type | Applies To | Impact | Severity | Patch Source | Chassis Type | # Installed | # Required | # Total | % Required | Link | Criteria |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2020-06-09 | 15 | KB4561603 | IE Cumulative | | Remote Code Execution | Critical | dit_server_wsus | server | 242 | 333 | 575 | 57,91 | +Info | (3) |
| 2020-06-09 | 15 | KB4561666 | Monthly Rollup | | Remote Code Execution | Critical | dit_server_wsus | server | 242 | 321 | 563 | 57,02 | +Info | (3) |
| 2020-06-09 | 15 | KB4561612 | Monthly Rollup | Windows Server 2012 | Remote Code Execution | Critical | dit_server_wsus | server | 8 | 120 | 128 | 93,75 | +Info | (3) |

RoboCISO conversation

This view was created according to the following criteria:

For a total number of assets > 100:

- (1) critical patches released more than 28 days ago that are still required in more than 10% of the assets OR
- (2) critical patches released more than 21 days ago that are still required in more than 25% of the assets OR
- (3) critical patches released more than 14 days ago that are still required in more than 50% of the assets

Figura C.4: Exemplo de *e-mail* do *Patching Alert* caso existam atrasos a reportar.

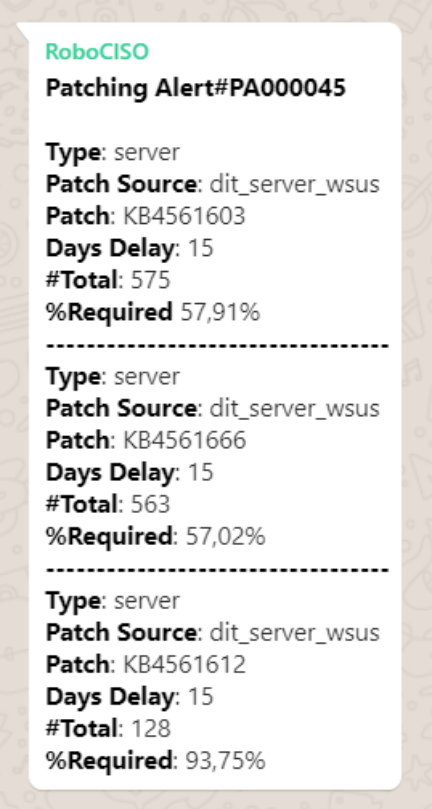



Figura C.5: Exemplo de mensagem do *Patching Alert* caso existam atrasos a reportar.

## C.3 DDoS Alert

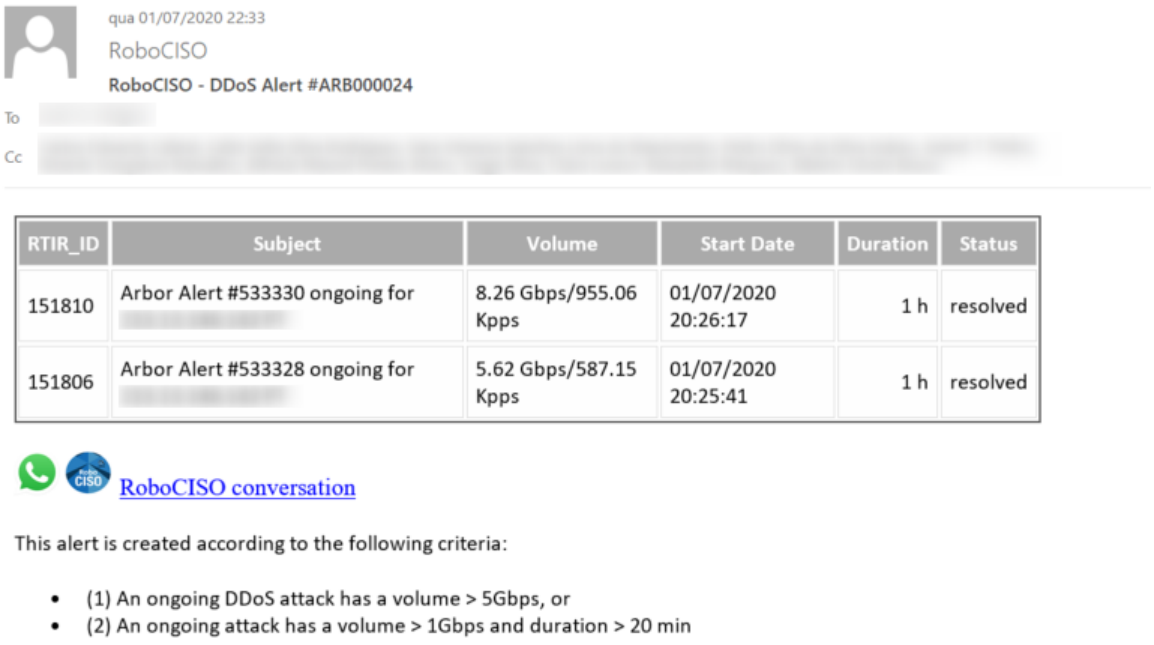



Figura C.6: Exemplo de *e-mail* do *DDoS Alert*.

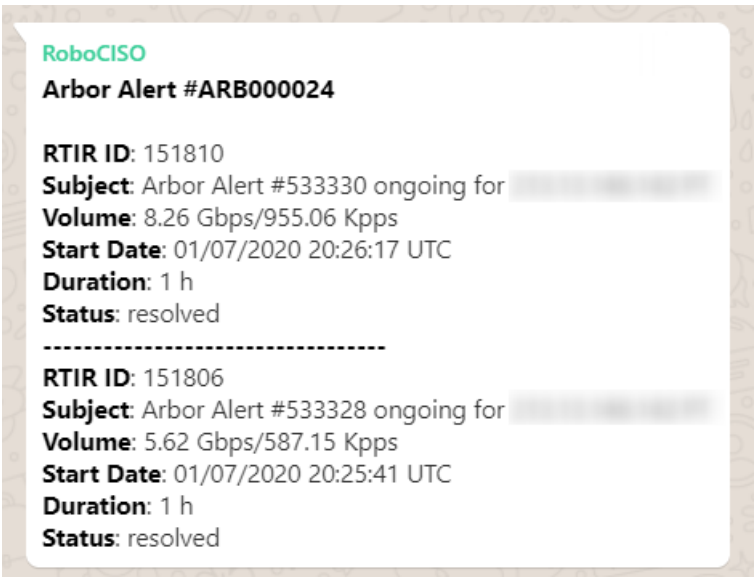



Figura C.7: Exemplo de mensagem do *DDoS Alert*.

## C.4 SLA Alert


| RTIR ID | Subject | Created Date | Classification | SLA | Duration | Status |
|---|---|---|---|---|---|---|
| 138479 | Arbor Alert #531590 done for PT | 26/06/2020 08:59:27 | | 4h | 73h | resolved |
| 138933 | FW: [CNCS\|CERT.PT #63953] [TLP: AMBER] Intrusão - Compromisso de Conta Não Privilegiada [MatterMost - PT] | 26/06/2020 13:43:33 | Account Compromise | 4h | 68h | resolved |

Figura C.8: Exemplo de *e-mail* do *SLA Exceeded*.

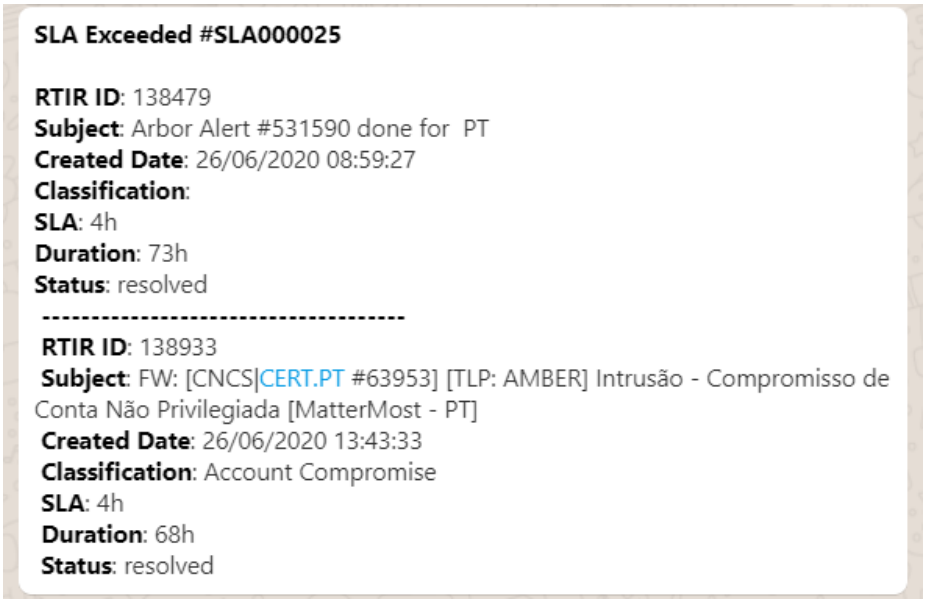



Figura C.9: Exemplo de mensagem do *SLA Exceeded*.

## C.5 Bitsight Alert

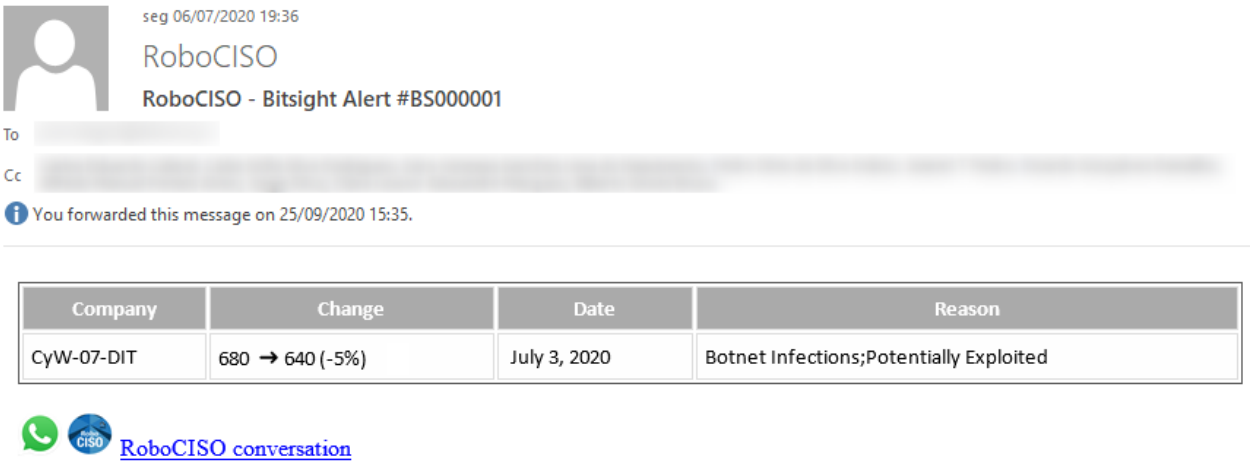



Figura C.10: Exemplo de *e-mail* do *Bitsight Alert*.

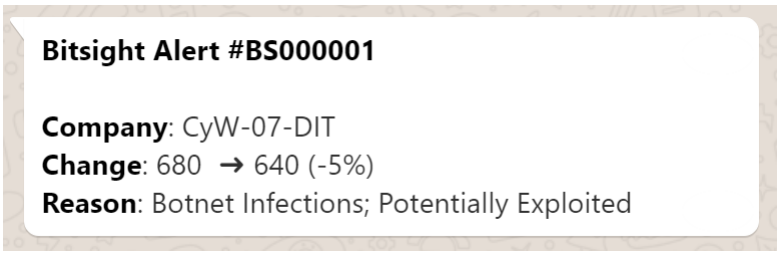
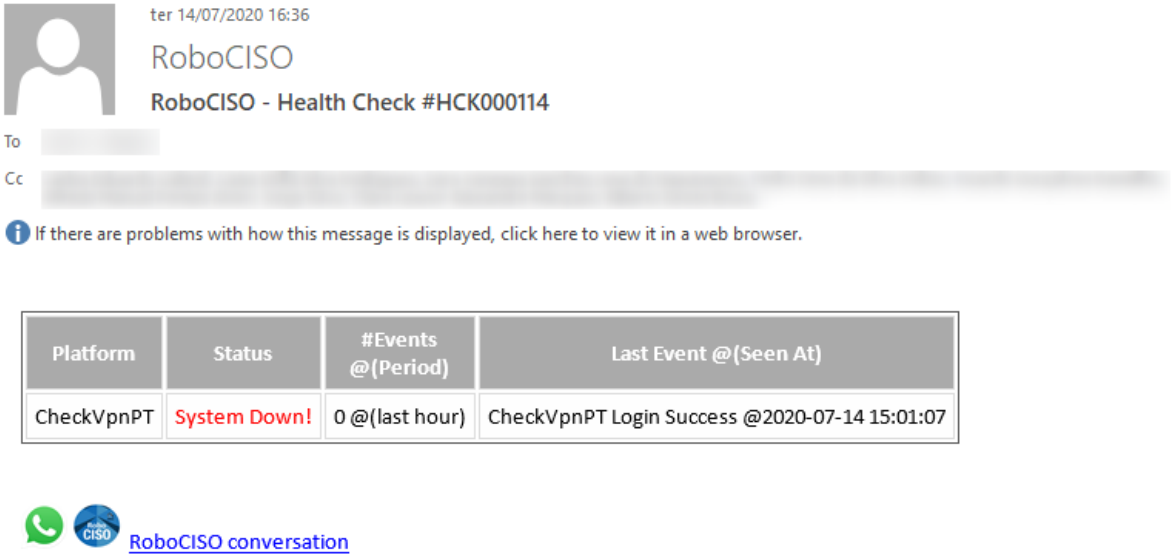



Figura C.11: Exemplo de mensagem do *Bitsight Alert*.

## C.6 Health Check



ter 14/07/2020 16:36

RoboCISO

**RoboCISO - Health Check #HCK000114**

To

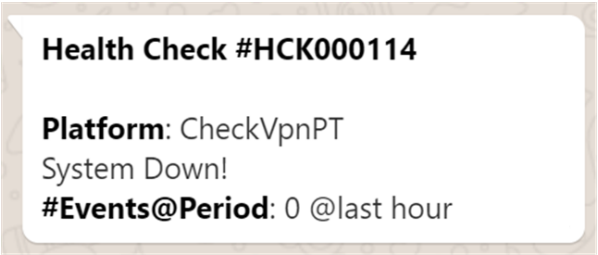
Cc

If there are problems with how this message is displayed, click here to view it in a web browser.

| Platform | Status | #Events @(Period) | Last Event @(Seen At) |
|---|---|---|---|
| CheckVpnPT | System Down! | 0 @(last hour) | CheckVpnPT Login Success @2020-07-14 15:01:07 |

RoboCISO conversation

Figura C.12: Exemplo de *e-mail* do *Health Check*.



**Health Check #HCK000114**

**Platform**: CheckVpnPT
System Down!
**#Events@Period**: 0 @last hour

Figura C.13: Exemplo de mensagem do *Health Check*.